IMPLEMENTATION OF A MACRO-PROCESSOR FOR STRING HANDLING

by

Constantin N. Stathopoulos

Graduate in Mathematics
University of Athens

A thesis presented for the
degree of Master of Science
of the
University of St. Andrews :
October 1971

Declaration

I declare that the following thesis is a record of research work carried out by me, that the thesis is my own composition, and that it has not been presented in application for a higher degree previously.

Constantin Stathopoulos

Certificate

I certify that Constantin N. STATHOPOULOS, graduate in Mathematics of Athens University, has spent four terms as a research student in the Computing Laboratory of the United College of St. Salvator and St. Leonard in the University of St. Andrews, that he has fulfilled the conditions of Ordinance 51 (St. Andrews), and that he is qualified to submit the accompanying thesis in application for the degree of Master of Science.

A.J. Cole
Supervisor

ACKNOWLEDGEMENTS

I should like to express my gratitude to my supervisor, Professor A.J. Cole, Director of the Computing Laboratory in the University of St. Andrews, who first suggested the subject of the following thesis, who was always willing to discuss and to advise and who read and commented on the original manuscript.

I am indebted, too, to the other members of the Computing Laboratory staff for their help during my work on this project.

C.N.S.

To

my parents

LIST OF CONTENTS

PREFACE

In this work TRAC is implemented in FORTRAN IV. This will enable various users to compile this FORTRAN version and use it in their own installations, making only minor modifications to meet individual specifications.

The way of the implementation allows adding either existing TRAC functions which are not included in this work or completely new, primitive functions needed for specific, well-defined purposes.

TRAC is a very flexible interactive language with versatile capabilities at execution time. The presented processor is programmed in FORTRAN IV using IBM 360 44PS and RAX facilities. It is compiled and intended to be used as a software package providing TRAC language facilities for the 360 RAX REMOTE ENTRY COMPUTING SYSTEM. It can be used under 44PS for special purposes.

TRAC is a member of the set 'STRING MANIPULATION LANGUAGES'. A thorough examination of this set helps to understand the basics of operations and techniques for dealing with strings. Another member of the same set is described briefly: this is the 'SNOBOL LANGUAGE'.

'String manipulation languages' is a subset of the set 'SYMBOL MANIPULATION LANGUAGES'. Some of the fundamental ideas and principles for symbol manipulation are included. The most-known languages, techniques and applications are mentioned, followed by references to allow further research and investigation.

The above are introduced in the following order:-

1. Symbol Manipulation Languages.
2. String Manipulation Languages.
3. TRAC Language.

PART I:

SYMBOL MANIPULATION

I. SYMBOL MANIPULATION

I.1 Introduction

Digital computers were first used as number processors. The applications mainly in scientific computation and data processing involved mostly numerical problems. Input data were interpreted as numerical values and internal calculations involved mathematical operations. For arithmetic computation there is a restriction of data to fixed or floating point and single- or double-precision numbers.

With the expansion of computer applications into new areas such as algebraic formula manipulation, information retrieval, computational linguistics, automatic decision making, translators and compilers, theorem proving, construction of bibliographies and indexes and others the ability to manipulate symbolic rather than numeric data became important in programming and soon it was realized that it would be more convenient to consider the computer mainly as a symbol processor transforming symbolic data into other symbols or structures of symbols.

Symbol manipulation consists of all the processes and techniques used in non-arithmetic computation, and employs the non-arithmetic instructions of a computer and operates on symbolic data producing usually symbolic results. Since the data are symbolic, there is a great variety of data forms.

I.2 Data

The computer is used as a tool to process data and transform them into results. Symbol manipulation refers to the processes for

manipulating symbolic data and special symbol-manipulation languages provide facilities for such manipulation.

To solve a problem with a digital computer it is necessary:-

(i) to encode the parameters of the real problem in one of several forms of symbolic data;

(ii) to interpret internally this symbolic data structure (the appropriate choice for the particular set of data makes the performance more efficient);

(iii) to transform these data into machine outputs with an appropriate computer program using a suitable symbol manipulation language;

(iv) to interpret the output as results significant to the particular problem.

I.2.1 Data structures

A data structure is defined as a set of rules and constraints which show the relationships that exist between individual pieces of data. Any information contained in these data is independent of their structure. An element of data may itself be another data structure. In this way it is possible to build data structures in various levels.

The most typical data structures for symbolic data are: Strings, Arrays, Queues and Stacks, Tables, Trees and Directed Graphs.

(1) String: A string is a sequence of characters from a well-defined set of characters or basic elements. It is an ordered sequence where each element is connected with its neighbours. For this reason to access a particular element in a string a sequential search is needed with one of its ends as starting-point.

(2) Arrays: An array is a set of elements with an associated set of integers, which defines the position of each element of the array. If the ordered set of integers has length m then the array is called m-dimensional and the individual integers subscripts.

(3) Queues and Stacks: These data structures can change dynamically. Both contain an ordered set of items and every new item is added at the end. Their difference is in accessing an item: in stacks only the last added item is accessible from the end of the stack, but in queues only the first added item is accessible from the front of the queue.

(4) Tables: A table consists of a set of items. Each item contains a key for identification together with other information associated with this item. An item can be accessed from a table by presenting its key. A new entry can be created by presenting the key of the added item together with its associated information.

(5) Tree: A tree is a set of nodes. A node may contain information and pointers to lower-level nodes. At the lowest level of the tree the node pointers point to structures external to the tree. Each tree has a topmost node which has no pointers from other nodes to it. This node is usually called the root of the tree. No node can have pointers to a previously-defined node. Consequently there is a unique way from the root to each node of the tree.

(6) Directed graphs: A directed graph is an extension of a tree. Each node can have pointers even to previously-defined nodes. Now there is not a unique way from the root to each node. It is usual to mark the direction on a line joining two nodes. In directed graphs it is possible to have a node pointing back to itself.

I.2.2 Storage structures

The computer storage consists of a set of ordered words. This is how computer storage is organized and this is the only structure that computer memories have. It is possible though by the use of programs and subroutines to create higher-level storage structures and make the use of computer storage more efficient. To the user the computer

itself appears to have this internal storage structure.

These storage structures are designed to match data structures or to provide an internal structure into which the data structure can easily be mapped. All these structures have been influenced by the serious burden of the basic structure of the computer storage into an ordered set of words. The most typical structures of computer storage are Vectors, Lists and Plexes.

(1) Vectors: A vector is the name given to a structure similar to the basic structure of computer storage. A vector is a set of elements, where elements are actual pieces of storage having the same length which can vary from a single bit of a word to a number of computer words.

The vector is completely defined when the address of its first element, the size of its element and its length are given. There is direct access to each element when the vector is defined as above and the order of this element in the array is given.

There is a one by one correspondence between a vector and the one-dimensional array. For this reason such an array without any change is mapped into a vector.

(2) Lists: A list is a sequence of elements with two fields each. The second field contains a pointer to the next element. The first field contains a pointer to the actual information defined by this element. This could be another list or some external data structure. The external data structure is called an atom and is assumed to be indivisible as far as the operations on the list are concerned.

To access a list a pointer to its first element is needed. The last element of a list contains the null pointer in the second field or a pointer to the first element in the case of a circular list. Using lists implies that there is no need any longer to have consecutive elements successively in store. This is a way for using the internal

storage more efficiently.

(3) Multi-linked lists: A multi-linked list is an extension of a list. It is a more flexible way which allows more efficient representation of complex data structures such as trees and directed graphs. This kind of structure was first given the name of "plex". A plex consists of a set of elements where each element is a vector of computer storage.

A plex is an extension of the list structure. The list element, usually 2-words, is expanded to a vector of K-words. This vector is divided into fields containing information or pointers to other vectors in computer storage.

I.2.3 Representation of data structures

When a form of data is given it must be related to the representation of that form within the computer memory. In addition to the data form a set of operations on these data is also given. These operations will be executed more efficiently if the appropriate internal representation is chosen. Some of the best-known ways of mapping data into internal storage follow.

The conventional way of storing a string is as a vector. Sometimes, however, lists are used for string representation to allow certain string operations to be performed more efficiently.

The most common way to store arrays is by using vectors and storing the array in consecutive elements of the vector. The order in which array elements are stored is according to their subscripts, usually with the first subscript varying most rapidly.

A stack is represented by a vector. A pointer is set to the top element of the stack. The length of the vector usually is the maximum expected length for the stack.

A queue is represented by a list. If it were represented as a

vector, each time an item was removed from the queue, it would result either in moving the elements down the vector to fill up the gap made at the base or alternatively the set of elements in the queue would gradually move down the vector. The best way is to represent the queue as a circular list with the pointer to the list pointing to the last added element.

Trees and directed graphs can be represented by lists but since there is an arbitrary number of pointers from each node the most efficient way is to use multi-linked lists.

Tables are usually represented by vectors. Lists and multi-linked lists may be used for appropriate kinds of tables, such as decision tables. The insertion or deletion associates the table entry with its key. According to the application there are direct access tables or tables where a searching function is carried out for finding the element given its key.

I.3 Symbol Manipulation Languages

With the increasing complexity in symbolic manipulations, it became evident that programming in machine-oriented languages would be tedious and time-consuming. Various techniques have been developed for handling symbolic data. As a result, a great variety of symbol manipulation languages were developed which when used with the appropriate data are powerful and efficient tools for dealing with symbolic manipulation.

As the applications in symbolic manipulation were expanded, some of these languages were used for problems that their implementors could not suspect at the time they designed them. This set of symbol manipulation languages is divided into subsets according to common characteristics of their members (languages).

I.3.1 List-processing languages

These are the first and more commonly used symbol manipulation languages. They include facilities such as copying lists, testing the equivalence of list structures, modifying the order of lists so as to insert, delete or replace elements. They do garbage collection and allocate free storage when needed. Usually they provide recursion facilities for list structures.

The oldest members of this set are IPL-V and LISP 1.5. IPL-V [6],[7] is a straightforward, low-level, list-processing language. LISP 1.5 [8],[9] is a more sophisticated automatic language which can be applied to a wide variety of non-numerical, mixed numerical and symbolic data-processing problems.

I.3.2 General purpose languages

These languages have built-in facilities for both symbolic and numerical computation. The best-known members of the set are SLIP, DYSTAL, LISP2 and FORMULA ALGOL.

A common characteristic in SLIP and DYSTAL is that both provide list-processing facilities for algebraic languages. LISP2 and FORMULA ALGOL provide facilities for symbolic and numeric computation. SLIP [10] has been used successfully for various list-processing applications. DYSTAL [11] is more suitable for arithmetic computations with a limited amount of list-processing.

LISP2 [12] is designed for a very large computer in a time-sharing environment. It is efficient for list manipulation and string processing. FORMULA ALGOL [13] provides built-in packages for formula manipulation. It is practical for medium-sized batch-processing computers but has limited processing facilities.

I.3.3 Linked-block languages

In the above-mentioned languages computer memory is assigned automatically and this does not concern the user. In the so-called linked-block languages the user is allowed a low-level control and responsibility for reserving and allocating the internal storage.

In this way it is possible for the user to define during the memory allocation various factors such as size of blocks in sequential storage and the locations of pointers to link these blocks. Thus the sophisticated programmer has the means to construct very efficient list-processing programs to fit his particular problems.

The best-known members of this set are L^6 [14] and CORAL [15]. In the L^6 language the linkage of elements is specified directly by the programmer who operates on them individually. L^6 provides rather few facilities but it is easier implemented and it provides bases for more sophisticated languages. CORAL is a higher-level list-processing language capable of more complex computation.

I.3.4 Algebraic-formula languages

These languages allow the computer to be used efficiently for various non-numerical mathematical problems. They are designed mainly for manipulating symbolic algebraic expressions. They are efficient in dealing with complex algebraic or analytic derivations on symbolic mathematical expressions.

The best-known members of this set are FORMAC, ALTRAN and AMBIT.

FORMAC and ALTRAN are designed mainly for manipulation of algebraic expressions. AMBIT [20], [21] is more general with a close resemblance to string manipulating languages. However, it is capable of operating on formal mathematical expressions. AMBIT would be a great help in the writing of a formula manipulator but it is not one. It is more useful for designing a formula manipulating language than

actual use as such. It can be used for string manipulations wherever matching of parenthesized symbols is involved.

ALTRAN [18],[19] is an efficient language for the problems for which it was designed. It uses both time and storage in a very efficient way.

FORMAC [16],[17] is a very flexible language which provides the user with both low-level functions which allow him to program his simple formula manipulation and high-level functions which carry out very complex operations.

I.4 String Manipulation Languages

Since the aim in the present thesis is to present the TRAC language and its implementation a more thorough examination of string manipulating languages follows.

I.4.1 Basic definitions and procedures in string manipulation

A string is a sequence of characters. When a string is defined explicitly it is usually enclosed in quotation marks. For example, "THIS IS A STRING", "A STRING", are two strings. When a string is defined in this way by value it is called literal. A string can have a name. Thus it would be possible, in any string manipulating language, to give the first string a name, e.g. STRING1, and the second STRING2, and to refer to both by name or by value.

Basic operations which a string manipulation language should provide are the following:

Concatenation: Link together two or more strings.

Deconcatenation: This is the opposite of concatenation, the ability to break a string into substrings. The most primitive way would be to take a string and break it into two substrings, one consisting of its first character and the other of the remaining characters. In all string manipulating languages such construction operations are provided usually in a more sophisticated way.

Branching: In addition to these operations a branch on equality

should be provided to allow conditional operations. This test on equality refers to the equality of characters and provides alternative transfers of control, or construction of different strings or execution of different procedures depending on the true or false value of the condition tested.

Naming facility: This enables the user to refer to a string or to parts of a string by name. There are times when it is inconvenient to refer to a string as a literal.

Procedure: This is the equivalent of a subroutine in programming. To be able to build more complex operations from the previous operations the language must provide means of composing, using and calling them as subroutines or subprograms.

Other basic operations are string matching and replacement. Facilities for integer arithmetic, referencing and input-output should preferably be included when designing the language.

There is a nearly unlimited number of operations and procedures that a language can provide. Here an important problem arises. How many of these features should an implementor include in this language to make it more efficient? The implementor must weigh both the advantages and disadvantages of increasing the power of the language by the provision of more operations and procedures, some of which may not be needed by the ordinary user.

An essential characteristic of a language is efficiency in certain types of problems. Efficiency is usually measured in time and storage requirements, two features in conflict with each other.

Finally a choice has to be made to use a compiler or an interpreter. An interpreter needs more time, but it allows more protection, being able to detect faults in the code presented by the user and to give messages for correction at the time when these faults occur.

Taking all these factors into consideration it is desirable when

designing a language to make it general for its possible applications and to allow the user to define from this language his own language in order to handle more efficiently his individual problems.

I.4.2 Pattern-directed languages

The pattern-directed string-processing languages allow operations on strings and transformations of these strings. Their structure is close to a mathematical processing model, the Markov algorithm.

The data string is compared with a pattern. If there is a match, a transformation of the data input data string occurs according to a format which is associated with that pattern. If the match is unsuccessful, there is no transformation.

The best-known members of this set are COMIT, SNOBOL, PANON and AXLE.

COMIT [22] has flexible input-output and subroutine-linkage facilities, a list facility for quick dictionary searching and logical subscripts.

SNOBOL provides names for strings, explicit function calls and automatic check for parenthesized strings.

PANON [23] provides power for recursive scanning of highly-structured string data.

AXLE [24] is similar to PANON. It includes certain rules defining the syntax of accepted strings.

I.4.3 Pattern-directed structure languages

These are more general symbol-manipulating languages with pattern-directed string-processing capabilities added to them. The result is a much more powerful system, since it provides facilities for both symbol manipulation and string processing.

The best-known members of this set are CONVERT, FLIP and COGENT.

CONVERT [25] and FLIP [26] are close to pattern-directed languages which operate on LISP structure. They are both flexible systems capable of using symbol and string manipulation techniques. COGENT [27],[28] has a special purpose, the processing of sentences of some context-free phrase-structure language.

I.4.4 Text and macro-handling languages

Two of the best-known such languages are GPM and TRAC.

The GPM [29] or General Purpose Macrogenerator is a symbol string processor with strings as input and output. It uses a substitution operation which is completely general and thus can be applied anywhere. It is a powerful system, including recursive functions and conditional expressions which can be implemented with only a few instructions.

TRAC will be described fully in Part II. It is similar in many respects to GPM but there is a major difference between the two. GPM allows the names of both system macros and programmer-defined macros to occur in the first argument position, while TRAC reserves the first argument position for names of system macros and requires programmer-defined macros to be specified in the second argument position.

I.5 SNOBOL Language [30],[31],[32]

I.5.1 Introduction

A brief description of the SNOBOL language is included. This language, together with the TRAC language, which will be fully described, will help to make clear most of the fundamental principles and techniques in string processing.

SNOBOL, which basically is a problem-oriented language, has been developed mainly to make effioient string manipulation with the computer.

Main sources, for designing and implementing the SNOBOL language, were other already existing languages such as COMIT and SCL. There are several versions of the language in use in various installations. A basic subset of the language is described.

The aim in designing the language was its simplicity and intuitiveness. Some of its fundamental string operations are creation of strings and alteration of strings depending on their contents.

I.5.2 Description of the language : The syntax is informally described by a metalinguistic language based on Backus-Naur form. It is not complete but gives the basic principles involved.

I.5.2.1 Syntax

1. <string symbol > :: = < letter> / <digit> / < special character >
2. <letter> :: = A/B/C/D/E/F/G/H/I/J/K/L/M/N/O/P/Q/R/S/T/U/V/W/X/Y/Z
3. <digit> :: = 1/2/3/4/5/6/7/8/9/0
4. <special character> :: = +/-/*/•/./$/(/)/'/b/=
5. <string> :: = ' < any sequence of string symbols not containing quotes > '
6. <string name> :: = < any sequence of letters, digits and •>
7. < unsigned integer > :: = < digit> /< unsigned integer >< digit>
8. < integer > :: = < unsigned integer> /+ < unsigned integer > /-<unsigned integer >
9. < string naming> / < string replacement> :: =
   < string name> b = b < string > / < string name > b̆ = b̆ < concatenated string> / < string name > b = b̆ < string name >
10. < concatenated string > :: = < any sequence of strings or string names separated by b >
11. < pattern matching > :: = < string name > b < string> / < string name > b < concatenated string >
12. < string variable > :: = < arbitrary variable > / < fixed-length

variable> /< balanced variable > /< mixed variable> /< nameless

variable> /< nameless of fixed length> /< nameless and balanced >

13. < arbitrary variable > :: = * < string name > *

< fixed-length variable> :: = * < string name > /N*

N = '< unsigned integer >'

< balanced variable > :: = *(< string name>)*

< mixed variable > :: = *(< string name > /N)*

< nameless variable > :: = **

< nameless of fixed length > :: = */N*

< nameless and balanced> :: = *( )*

14. < pattern matching and replacement > :: =

< string name > b < string > b = b < string> /

< string name > b < string > b = b < concatenated string > /

< string name > b < concatenated string > b = b <string> /

< string name > b < concatenated string > b = b < concatenated string >

15. < snobol program > :: = < ordered sequence of snobol statements >

< snobol statement> :: = (< label>) b < rule > b (< goto >)

< label > :: = <digit > < any string symbol except b > /

<letter > < any string symbol except b >

< first statement character > :: = b/ <letter > / < digit> /·/*

< rule > :: = <string reference> b (< pattern >) b = b (< replacement >)

or < rule > :: = <pattern matching and replacement >

< goto > :: = /S(< label >)/F(< label>)/S(< label >)F(< label >)/

/(< label >)

I.5.2.2 Examples

5. Strings:- 'THIS IS A STRING'

'THE SUN IS WARM, THE SKY IS CLEAR, THE WAVES ARE

DANCING FAST AND BRIGHT'

'++//--**.....,£££££ ('') bbbbbbbbbb'

'123+321*100/A+B.C=STRING'

'1,2,3,4,5,6,7,8,9,0,A,B,C,D,E,F,G,H,I,J'

6. String names:- VERSE, VERSE.10, 30.ABC, RULE.10, STRING.NAME1, FIRST.NAME

9. String naming and replacement:-

/I VERSE.1 = 'THANKS TO ITS TENDERNESS, ITS JOYS, AND FEARS'

/II VERSE.2 = 'TO ME THE MEANEST FLOWER THAT BLOWS CAN GIVE'

/III VERSE.3 = 'THOUGHTS THAT DO OFTEN LIE TOO DEEP FOR TEARS'

/IV POEM = VERSE.1 '***** TO ME THE MEANEST FLOWER THAT BLOWS CAN GIVE *****' VERSE.3

/V VERSE.2 = VERSE.3

/VI POEM = 'VERSE.2=' VERSE.2 '*****VERSE.3=' VERSE.3

11. Pattern matching:- VERSE.1 ITS, VERSE.3 OFTEN LIE, VERSE.2 E, POEM VERSE.2 '*****VERSE.3=', POEM VERSE.1

12-13. String variable:-

/I VERSE.1 'TENDERNESS' *VARBL.1* 'AND FEARS'

/II VERSE.2 'OFTEN b' *NEXT.WORD* 'b'

/III VERSE.1 *VARIABLE/'6'*

/IV VERSE.2 *X/N* *REMAINING.STRING* N = '9'

/V VERSE.1 'ITS' *VAR* 'FEARS'

/VI VERSE.1 'ITS' ** 'FEARS'

/VII VERSE.2 *X/'9'* *Y/'5'* *Z*

/VIII VERSE.2 *X/'9'* * /'5'* *Z*

/IX X = '(A+B+C)' X *(X1)* X1 = (A+B+C)

/X Y = '(A-(B-C)+D)' Y '('*(Y1)*')' Y1 = A-(B-C)+D

14. Pattern matching and replacement:-

```
/I   VERSE·1  'TENDERNESS' = 'MATCH'

/II  VERSE·3  'TEARS'      = 'TEARS'  VERSE·4

   where VERSE·4 = '***** VERSE·4 = THE CLOUDS THAT GATHER

   ROUND THE SETTING SUN'

/III VERSE·3  'TEARS' VERSE·4 = 'TEARS'

/IV  POEM     'VERSE·2=' VERSE·2 = 'VERSE·2 = TO ME THE MEANEST

   FLOWER THAT BLOWS CAN GIVE'

/V   POEM     'VERSE·2 = TO ME THE MEANEST FLOWER THAT BLOWS CAN

   GIVE ***** VERSE·3=' VERSE·3 = VERSE·1 '* TO ME THE MEANEST

   FLOWER THAT BLOWS CAN GIVE *'
```

15. SNOBOL program:-

```
BEGIN  INTEGER                  = '0,1,2,3,4,5,6,7,8,9'

NUM-1  INTEGER          *I*',' =            /F(END)

NUM-2  ALPHANUMERIC·TEXT     I =            /S(NUM-2)F(NUM-1)

END    BEGIN
```

I.5.2.3 Semantics

1-9:- The character set consists of 48 string symbols. All these symbols are considered as just symbolic characters without any concern to what they actually represent: for example,- does not necessarily imply subtraction.

A string is the basic data structure of the language and can contain any combination of string symbols. It is sometimes referred to as a literal string and is always enclosed between quotation marks. The length of a string is a number which denotes how many elements the string has. This length is 0 when the string has no elements at all and then it is called the null string.

The language provides facilities for referring to a string not only by its contents, but also by its name, since it is possible to

assign one to it. The name of a string may contain any sequence of letters or digits and periods.

9. String naming and replacement:- With the naming facility the name string which is on the left of the equals sign is assigned to the literal string which is on the right. VERSE·1, VERSE·2, VERSE·3 are the names assigned to three lines of a poem. Thus it is possible to have two equivalent references to the same line by name, e.g. VERSE·2, or by contents, e.g. 'TO ME THE MEANEST FLOWER THAT BLOWS CAN GIVE'.

The string on the right may be a concatenated string. Such a string is formed by taking a sequence of strings and/or string names as one string. In the example 9/IV the content of the string POEM is formed by concatenation of various strings and string names. In fact POEM has as content the string 'THANKS TO ITS TENDERNESS, ITS JOYS, AND FEARS ***** TO ME THE MEANEST FLOWER THAT BLOWS CAN GIVE ***** THOUGHTS THAT DO OFTEN LIE TOO DEEP FOR TEARS'.

The equals sign has, as usual, a special meaning. It is the replacement operator. Naming might be considered a special case of the more general one which is replacement. Replacement here refers to string replacement. It means substitution of the contents of the string which is on the left with the contents of the string which is on the right of the equals sign. Of these two strings the former is replaced but the latter remains with the same contents.

After execution of the examples 9/V and 9/VI the strings VERSE·2 and POEM have as content respectively 'THOUGHTS THAT DO OFTEN LIE TOO DEEP FOR TEARS', 'VERSE·2 = THOUGHTS THAT DO OFTEN LIE TOO DEEP FOR TEARS ***** VERSE·3 = THOUGHTS THAT DO OFTEN LIE TOO DEEP FOR TEARS'. Since the previous replacement has taken place both VERSE·2 and VERSE·3 have the same content.

11. Pattern matching:- This facility makes it possible to specify if a sequence of characters, the "pattern", exists in a given string.

The action taken is the scanning of the given string from left to right for a match with the given pattern. In fact a string is searched to find out if a particular substring is contained or not.

The pattern may be a string, a string name or a concatenated string and the match successful or unsuccessful. In brief we can say that the string named on the left is to be searched for the pattern specified on its right. The success or failure of matching may be used for control of subsequent operations.

In example 11 the strings VERSE·1, VERSE·3, VERSE·2, POEM, POEM would be scanned for matching with the corresponding patterns ITS, OFTEN LIE, E, VERSE·2 '***** VERSE·3=', VERSE·1. All matches except the last one are successful. The last is unsuccessful since there is no VERSE·1 in the string POEM in its last definition in example 9/VI.

12-13. String variables:- The pattern-matching previously described presents some serious limitations. It cannot be applied to a pattern consisting of two substrings with some irrelevant characters in between. The string variable provides the facility to deal with this kind of matching.

Consider the example:

VERSE·1 'TENDERNESS' *VARBL·1* 'AND FEARS'

The content of VERSE·1 is scanned from left to right for matching the substring patterns 'TENDERNESS', 'AND FEARS' with some irrelevant characters in between. The string name VARBL·1 enclosed in a pair of asterisks represents the string variable which during the matching procedure takes care of the irrelevant characters between the two patterns. When the match is successful the irrelevant characters form the content of the string variable. A by-product of this example would be the formation of a new string VARBL·1 = ', ITS JOYS'. The string thus formed, VARBL·1, is completely independent of the string VERSE·1 and it would keep its original content in the event of an unsuccessful

match.

Example 12/II makes it possible to find the next word after the word OFTEN. The string formed is NEXT·WORD = 'LIE'. The variable described is an arbitrary string variable. There are various string variables. The fixed-length variable allows matching only with a substring of specified length. This variable is indicated by adding after the variable name a slash followed by the string length which may be a number or string name which has numeric contents. Two of the applications of this variable are demonstrated in examples 12/III and 12/IV, where the strings formed are VARIABLE = 'THANKS', X = 'THOUGHTS', REMAINING·STRING = 'THAT DO OFTEN LIE TOO DEEP FOR TEARS'. This is a way of deconcatenation that is to decompose one string into one or more substrings specifying the length for the substrings, without any change in the content of the original substring.

The nameless string variable is introduced when the string variable is not really needed but is used only to help to match various patterns which are not consecutive. Examples 12/V and 12/VI demonstrate this possibility. In the 12/VI example the nameless variable has the same content as the string variable VAR, that is 'TENDERNESS, ITS JOYS, AND'. When this variable is not needed for further manipulation, the nameless variable should be used.

The nameless string variable of fixed length is used for just skipping a certain number of symbols in a pattern-matching. In examples 12/VII and 12/VIII, the former results in the substrings X = 'THOUGHTS', Y = 'THAT ' Z = 'DO OFTEN LIE TOO DEEP FOR TEARS'; the latter results in X = 'THOUGHTS', 5 successive characters skipped, Z = 'DO OFTEN LIE TOO DEEP FOR TEARS'.

The balanced string variable is used mainly in applications which involve algebraic manipulation of symbolic mathematical expressions, where parentheses are of primary importance usually occurring in pairs.

A balanced string variable can only match a substring which is balanced with respect to parentheses. A balanced string variable has its name enclosed between a pair of parentheses. Examples 12/IX and 12/X demonstrate two applications of balanced string variables X1 and Y1. Example 12/X in particular demonstrates a convenient way to strip off the outer pair of parentheses of an expression without changing the content between this pair of parentheses.

14. Pattern matching and replacement:- This is one of the most important facilities that the SNOBOL language provides. It allows the alteration of the contents of a string, which is one of the basic concepts in string manipulation.

In example 14/I the string VERSE·1 is scanned from left to right for a match with the pattern 'TENDERNESS'. When the match is successful, the characters of the string which match the pattern are replaced by the string which is on the right of the equals sign. When the match is unsuccessful, the string remains unaltered. In example 14/I there is a match for the pattern and consequently an alteration of the string VERSE·1 = 'THANKS TO ITS MATCH, ITS JOYS AND FEARS'.

The pattern and/or its replacement may be any combination of strings, string names or string variables. When there is nothing following the equals sign, the characters matching the pattern in the string are deleted. For example:

VERSE·1 'TENDERNESS' =

would result in VERSE·1 = 'THANKS TO ITS, ITS JOYS, AND FEARS', with the matching characters TENDERNESS deleted from the string VERSE·1.

The altered strings of the other examples follow:-

Ex. 14/II VERSE·3 = 'THOUGHTS THAT DO OFTEN LIE TOO DEEP FOR TEARS

***** VERSE·4 = THE CLOUDS THAT GATHER ROUND THE SETTING SUN'

Ex. 14/III VERSE·3 = 'THOUGHTS THAT DO OFTEN LIE TOO DEEP FOR TEARS'

Ex. 14/IV POEM = 'VERSE·2 = TO ME THE MEANEST FLOWER THAT BLOWS CAN

GIVE ***** VERSE·3 = THOUGHTS THAT DO OFTEN LIE TOO DEEP FOR TEARS'.

Ex. 14/V POEM = 'THANKS TO ITS TENDERNESS, ITS JOYS, AND FEARS* TO ME THE MEANEST FLOWER THAT BLOWS CAN GIVE* THOUGHTS THAT DO OFTEN LIE TOO DEEP FOR TEARS'.

15. SNOBOL program:- A SNOBOL program is an ordered sequence of statements. These statements are executed in the given order unless a branching occurs and then this order changes.

A statement contains three parts separated by blanks. Not all of them are mandatory. The non-mandatory parts are enclosed between parentheses in the syntax description. The order in the statement parts is the following:

(i) A label which gives a name to the statement for subsequent reference.

(ii) A rule which may be decomposed into three parts, namely string reference, pattern and replacement.

(iii) A goto, which may provide a conditional branching to a labelled statement.

A label may contain any symbols except blanks but it must always begin with a digit or a letter. Any string name can be used as a label. In example 15 START, NUM-1, NUM-2, END are statement labels.

A rule may contain three parts, the first two being separated by at least one blank and the second two by an equals sign, which must also have at least one blank on either side. The order of these parts from left to right is:

(i) The string reference which denotes the string to be manipulated.

(ii) The pattern which is the left of the equals sign string.

(iii) The replacement which is the right of the equals sign string.

Only the string reference is mandatory; any other part may be absent depending on the rule and its application. Consider a rule from a

statement of example 15:

e.g. INTEGER *I* ',' =

Here the string reference is INTEGER, the pattern *I* ',', and the replacement is absent which means deletion of the matching characters.

A goto part of a statement begins with a slash followed by one or possibly more of the following parts:

(i) Unconditional transfer of the form (label) which after completion of the statement transfers control to the statement with the specified label.

(ii) Conditional transfer on failure, which has the form F(label). This transfers control, if the statement fails, to the statement with the specified label, otherwise the next statement is executed.

(iii) Conditional transfer on success of the form S(label) similar to (ii), with transfer of control on success.

Examples of goto are in example 15:

/F(END) is conditional transfer on failure to the statement END.

/S(NUM-2)F(NUM-1) is conditional transfer of control on success to the statement NUM-2 and on failure to the statement NUM-1.

A full statement of the same example is:

NUM-2 ALPHANUMERIC·TEXT 1 = /S(NUM-2)F(NUM-1)

NUM-2 is the label

ALPHANUMERIC·TEXT I = is the rule

ALPHANUMERIC·TEXT is the string reference

I is the pattern

blank is the replacement

/S(NUM-2)F(NUM-1) is the goto

NUM-2 is the label of the next executed statement on success

NUM-1 is the label of the next executed statement on failure.

The first character of a statement may be b blank for an unlabelled statement, letter or digit for the beginning of a label, · for

continuation of the preceding statement, * for a comment.

A SNOBOL program (see example 15) is an ordered sequence of statements with the last statement having the label END and a string reference which is identical with the label of the first executable statement (BEGIN in the example).

Statements are executed successively one after the other unless a goto specifies transfer of control. The execution terminates when control is transferred to the statement labelled END.

The execution of the program, in example 15, begins with the statement BEGIN which forms a string named INTEGER and containing all digits from 0 to 9. The next statement NUM-1 names the first digit in INTEGER to be I and deletes this digit and the following comma from the string INTEGER since the replacement string is blank. The next statement NUM-2 will be executed until the occurrence of a failure which will transfer control to the statement NUM-1. When a failure in this statement occurs control will be transferred to END and it will terminate execution.

When control is in statement NUM-2, the reference text ALPHANUMERIC·TEXT is scanned for the occurrence of the particular digit and if the match is successful the digit is deleted from the text which is previously defined and contains letters and digits. After the deletion of the digit, control is transferred again to NUM-2 statement. In fact this is a loop executed as many times as the number of occurrences of the digit examined in the text. When a failure occurs, there is transfer of control to the statement NUM-1 to select another digit until the string INTEGER becomes null and then control is transferred to END, the execution stops and the result of the whole process is the deletion of all digits from the given text.

I.5.3 Other features of the language

I.5.3.1 Back referencing

A string variable may be used to perform back referencing in a pattern-matching operation. Thus a pattern may contain a string name which is the same as the name of a variable previously used in the pattern. For example if EXPRESSION = '(X+Y) * (X+Y+Z/2) *-3/2(X-Y) -(X+Y)' the pattern-matching EXPRESSION '(' *VAR* ')' *REM* '('VAR')' would assign X+Y to VAR. Since the scanning proceeds from left to right, any attempt to match the string VAR will be made after assigning a temporary value to the variable *VAR*.

Another example, if VERSE = 'AND ANSWER, ECHOES, ANSWER, DYING, DYING, DYING' the pattern-matching VERSE * V /'6' * *OTHER* V will scan VERSE for the occurrence of 6 symbols which are repeated and will form the substrings V = 'ANSWER' and OTHER = ', ECHOES,'.

I.5.3.2 Indirectness

All references to string names can be thought of as direct address referencing. It is useful sometimes to provide facilities for indirect address referencing. This is known as Indirectness in SNOBOL and enables the user to refer to strings indirectly as well. An indirect reference to a string is made by adding the symbol $ in front of the string name. For example if STRING = 'ADDRESS 1000', writing $ STRING is equivalent to ADDRESS 1000.

An example of the utility of this feature is the following:

```
BEGIN   TEXT          *SIGN/'1'*  *REM*     /($ SIGN)
A       TEXT = REM
B       TEXT = REM    '-'
C       TEXT = REM    '*****'
```

The Text is scanned. The first symbol is assigned to SIGN, the remaining to REM. If this symbol is A, B or C the control is uncondition-

ally transferred to the statements labelled A, B or C. If another symbol is found the execution will terminate since a transfer is made to a non-existent statement.

I.5.3.3 Arithmetic

Simple arithmetic on signed or unsigned integers may be performed in SNOBOL. It is possible to assign a numeric value to a string by making its content a signed or unsigned integer. Operations such as addition, subtraction, multiplication, division and exponentiation may be performed on any two of these strings with corresponding symbols +, -, *, /, • with one blank at least before and after each arithmetic operator.

The following examples demonstrate how arithmetic is performed in SNOBOL:

```
V1 = '10'    V2 = '20'    V3 = '30'    V = '2'
W1 = V1 + V3    W2 = V1 * V2    W3 = V3 / V1    W4 = V2 / V3
W5 = V2 • V    W6 = V2 / '0'    W7 = '-40' / V1    W8 = '-40' / V3
W9 = A-B    W10 = V1 * V2 + V3.
```

The values assigned to W1, W2, W3, W4, W5 are '40', '200', '3', '0', '400'. W6 retains its content since it is impossible to divide by 0. W7 = '-4'. W8 = '-1'. W9 is not assigned a value since A-B is an illegal expression without the necessary blanks before and after the arithmetic operator. W10 is not assigned a value. The second part should be written with parentheses since only two operands are allowed in arithmetic expressions. The execution of the SNOBOL program stops when an illegal statement occurs as in the last two cases.

I.5.4 Input-output

Two commands •READ and •PRINT preceded by the string reference SYS are used for INPUT and OUTPUT operations.

The command •READ causes a string to be read from the INPUT device. If it is followed by a pattern a subsequent scan for matching with the pattern takes place. For example with the statement

```
DATA    SYS    •READ    *INPUT*    '0'
```

a string is read from the input device and gives the name INPUT to all characters before the first zero. If either there are no more data on the input device or the match with the pattern is unsuccessful the read statement fails.

The command •PRINT causes every string that follows to be printed on the output device. For example the statement

```
OUTPUT    SYS    •PRINT    'STRING='    STRING
```

would print the literal STRING= followed by the contents of the string named STRING.

I.5.5 Scanning algorithm

A pattern may consist of strings, string names or variables. The matching procedure is fully described by the following algorithm, which is quoted from the original paper on SNOBOL [30].

"Rule 1. An attempt is made to match the first pattern element starting at the first symbol of the string. If this match cannot be made, the match is attempted starting at the next symbol of the string, and so on.

Rule 2. The matching proceeds from left to right, successively matching pattern elements. Each pattern element matches the shortest possible substring.

Rule 3. If at some point an element cannot match a substring, an attempt is made to obtain a new match for the preceding pattern element. This new match is accomplished by extending the substring formerly matched to obtain the next shortest acceptable value. If this extension

can not be made, Rule 3 is applied again. If there is no preceding element a new match is attempted according to Rule 1.

Rule 4. If the last pattern element is an arbitrary string-variable (i.e. not fixed-length or balanced), its matching substring is extended to the end of the string.

The pattern match succeeds when the last pattern element has been matched. The pattern match fails when the first element cannot be matched."

The scanning algorithm is demonstrated in the following example:

```
EXPR  =  'A/(B+C+D)*((A+B)+C)'

EXPR     '('  *(E)*  ')'
```

Rule	Match	Pattern Element	String Element
1		(	A
1		(	/
1		(	(
2		*(E)*	B
2		)	+
3		*(E)*	B+
2		)	C
3		*(E)*	B+C
2		)	+
3		*(E)*	B+C+
2		)	D
3		*(E)*	B+C+D
2		)	)

E = B+C+D

I.6 References

[1]. B. Raphael, "Aspects and applications of symbol manipulation", Proc. 21st Natl. Conf. ACM (August 1966).

[2]. J.E. Sammet, "An annotated descriptor based bibliography on the use of computers for doing non-numerical mathematics", Computing Rev. 7, No. 4 (July-August 1966).

[3]. D.E. Knuth, The Art of Computer Programming, Vol. I: Fundamental Algorithms, Addison-Wesley Publishing Company, Inc. (1968).

[4]. J.A.N. Lee, The Anatomy of a Compiler, New York, Reinhold Publishing Corp. (1967).

[5]. A Comparative Study of Programming Languages (Macdonald Computer Monographs).

[6]. A. Newell (ed.), Information Processing Language-V Manual, Prentice Hall, Englewood Cliffs, N.J. (2nd edn. 1963).

[7]. A. Newell, "Documentation of IPL-V", Comm. ACM 6, No. 3 (March 1963).

[8]. J. McCarthy et al., LISP1·5 Programmer's Manual, MIT Press, Cambridge, Mass. (1962).

[9]. E.C. Berkeley and D.G. Bobrow (eds.), The Programming Language LISP: Its Operation and Applications, MIT Press, Cambridge, Mass. (1966).

[10]. J. Weizenbaum,"Symmetric list processor", Comm. ACM 6, No. 9 (September 1963).

[11]. J.M. Sakoda, DYSTAL Manual, Sociology Computer Laboratory, Brown University, Providence, R.I. (1965).

[12]. P.W. Abrahams et al., "The LISP2 programming language and system", AFIPS Proc. FJCC 29 (November 1966).

[13]. Formula Algol User's Manual, Carnegie Institute of Technology, Pittsburgh.

/14/. "A programmer's description of L^6", Proc. ACM 9, No. 8 (August 1966).

/15/. W.R. Sutherland, "The CORAL language and data structures", contained in Tech. Report 405, MIT Lincoln Laboratory, Lexington, Mass. (1966).

/16/. J.E. Sammet, Formula Manipulation by Computer, TR 00.1363, IBM Systems Development Division, Poughkeepsie, N.Y. (November 1965).

/17/. FORMAC Manual, IBM Corp., Program Information Dept., 40 Saw Mill River Road, Hawthorne, N.Y.

/18/. W.S. Brown, "A language and system for symbolic algebra on a digital computer", Proc. IBM Scientific Computing Symposium on Computer-Aided Experimentation (October 1965).

/19/. W.S. Brown et al., "The ALPAK system for non-numerical algebra on a digital computer", Bell System Tech. J. 42 (1963), 2081; 43 (1964), 785; 44(1964), 1547.

/20/. C. Christensen, "On the implementation of AMBIT, a language for symbol manipulation", Comm. ACM 9, No. 8 (August 1966).

/21/. C. Christensen, "Examples of symbol manipulation in the AMBIT programming language", Proc. 20th Natl. Conf. ACM (August 1965).

/22/. V.H. Yugve, COMIT Programming, MIT Press, Cambridge, Mass. (1966).

/23/. A. Caracciolo et al., "PANON-1B, a programming language for symbol manipulation", abstract in Comm. ACM 9, No. 8 (August 1966).

/24/. K. Cohen and J.H. Wegstein, "AXLE, an axiomatic language for string transformations", Comm. ACM 8, No. 11 (November 1965).

/25/. A. Guzman and H. McIntosh, "CONVERT", Comm. ACM 9, No. 8 (August 1966).

/26/. W. Teitelman, "FLIP, a format list processor", Memo MAC-M-263, MIT Project MAC, Cambridge, Mass. (1966).

/27/. J. Reynolds, COGENT Programming Manual, ANL-7022, Argonne National Laboratory (March 1965).

/28/. J. Reynolds, "An introduction to the COGENT programming system", Proc. 20th Natl. Conf. ACM (August 1965).

/29/. C. Strachey, "General-purpose macrogenerator", Computer J. (October 1965).

/30/. D.J. Farber et al., "SNOBOL, a string-manipulation language", J. ACM 11, No. 1 (January 1964).

/31/. "The SNOBOL3 programming language", Bell System Tech. J. (July-August 1966).

/32/. J.W. Backus, The Syntax and Semantics of the Proposed International Algebraic Language of the Zürich ACM-GAMM Conference, ICIP, Paris (June 1959).

PART IIA:

TRAC LANGUAGE:

DESCRIPTION OF TRAC

PART II: TRAC LANGUAGE

A. DESCRIPTION OF TRAC

IIA.1 Introduction

TRAC derives its name from "Text Reckoning and Compiling Language". It is designed to be a user's language providing efficient use and control of the computer and its back-up store, employing the reactive typewriter.

By a reactive typewriter is meant a teletypewriter, or just a typewriter with teletypewriter facilities, connected to a powerful computer system which allows multiple access time-shared operations. In the reactive typewriter system the user in front of the reactive typewriter keyboard is considered as the central point. The computer and its peripheral devices are used as peripheral service units to the reactive typewriter.

The TRAC language system was designed and developed as a basic software package for the reactive typewriter. This is a text or string manipulation language. It provides facilities for writing programs or procedures, which is the usual term for programs in string processing; for accepting, naming and storing any character string from the typewriter; for treating any string of characters at any time as an executable procedure or as name or as literal text; for printing out any character string.

The language has been formed with the aid of the programming concept, "macro", and its extension and generalization on symbolic strings. The TRAC language has the very important feature that it accepts, defines and stores procedures and thus extends indefinitely its

capabilities. It can handle effectively iterative and recursive procedures and use character strings, integers and Boolean vector variables.

IIA.2 Objectives of the TRAC Language

Most of the main goals in designing the TRAC language follow. Some of them were not evident from the beginning but appeared during the development of the language.

Since it is a string manipulation language it should provide facilities for the basic operations with strings, such as:-

- Accept as input any character or string of characters coming from the teletypewriter.
- Store them as text or literal or even as named strings after giving them a name.
- Operate on them or emit them according to the application.

All characters, even the control characters like carriage return, space, or the null character which means the absence of a character, may be included in the above operations. Each string can be treated as text or as a procedure or as a name.

Procedures can be produced by combining the primitive functions of the language. These procedures may be used for more efficient string processing. The language has great flexibility in dealing with strings. A string defined as text may be treated as a procedure or vice versa. For example, a previously defined procedure can be treated temporarily as a character string and updated and the corrected text used subsequently as a new updated procedure. Simple integer arithmetic, mainly for counting and other elementary operations, may be performed.

It provides for use of the back-up store such as tape, disc, drum.

Functions allow the user to store or retrieve named strings and to control the structure and organization of them on these mass storage devices.

TRAC is intended to be a user's program in a multiple-access time-shared environment and as such it is one of the many programs which are simultaneously connected with the computer. A problem for further investigation could be to include TRAC as part of the Operating System. In this way it could be shared by many users on a Real-Time basis.

TRAC is mainly a terminal language in the sense that the user communicates with the computer through the keyboard of a teletype or a video display unit. Trivial typing errors must not affect the system which should allow easy self-recovering. Some diagnostic messages could be advantageous for a user who could then follow closely the internal procedure.

Syntax is simple and unambiguous without priorities of rules or symbols. It is as general as possible and not biased towards specific applications. It contains only the necessary control characters and provides facilities for changing them under control of TRAC. The language is clear and elegant and can be learned very easily.

As a string manipulation language, it is mainly concerned with handling of strings. Strings are treated uniformly and atomic string structures are avoided as causing inconvenience in concatenation or deconcatenation of strings. The string length is arbitrary for easy string manipulation. Thus any string can be divided into others or any two or more strings can form another string. The most usual form of a string is to stand for itself as a text at the literal level, but the language provides facilities for treating strings also as names or procedures and the naming facility is used for indirect addressing, when needed.

It is possible to nest primitive functions to any depth, to perform iterations and recursions, with the only limitation being the available computer store, and to define and execute a procedure using other already defined procedures. Each procedure can act upon itself and upon others providing the system with powerful self-reference capabilities.

The set of primitive functions should be carefully selected. It should be wide enough to contain all necessary functions for easy string manipulation processing and should not require the user to define procedures even for common string operations. It should not be too detailed since then it charges the user for facilities he most likely does not need to use. An optimum number of functions would provide versatility and generality to the system.

TRAC being an interactive language, decisions might be taken, during the execution of procedures, which change and redefine procedures or data. Procedures are capable of defining and storing new strings by giving them unique names, and of using them either as text or procedure according to the particular application.

Another of the important decisions was that the input string typed by the user should be identical to the text which controls the TRAC processor internally. This implies that text and procedures are kept in memory exactly as they were typed. Whenever they redefine or create new procedures, these are also kept in the same form.

The TRAC interpreter, also known as TRAC processor, interprets this code into machine code and in this way the computer accepts and executes procedures written in the TRAC language. It is possible to display at any time any text or procedure in the same format as the one the TRAC processor acts upon during execution. It is also possible to stop execution and re-initialize the TRAC processor at any time without losing any of the strings already stored.

TRAC is called a homoiconic language since text and procedures are represented internally and externally in the same way. The modular construction and the uniform syntax of the language extend its capabilities for specialized applications to various installations. This is because of its ability to define new functions, similar to those existing, but applicable to specific kinds of operations.

IIA.3 Main Sources of TRAC Language

TRAC, like all the programming languages, has its sources of inspiration in earlier languages. It has been developed after a careful study of other problem-oriented languages. The ambition was to devise a language at least as powerful as they were but without their limitations. Amongst the languages, the most thoroughly examined were COMIT, LISP and IPL-V.

The major disadvantages of COMIT are:-

- The use of a compiler instead of an interpreter. This results in making impossible the modification of text or redefinition of procedures at execution time.
- Its line-by-line format and the inability of a procedure to call another, operate on it and produce some result.

In LISP the limitations are:-

- The present restrictions in the structure of its data due to the atomic symbols.
- It has a great orientation towards mathematical logic.
- It is not a language for easy manipulation from a keyboard.
- There is a certain amount of confusion due to a dual language, the M-language, read easily, which is used externally, and the S-language on which the LISP processor operates.

IPL-V appears closer to conventional computer programming than to a user's language for the reactive typewriter.

The fundamental ideas for TRAC came from three papers by McIlroy and colleagues describing a macro assembly system. These papers made clear the power of such a system. It was stressed that if such a system could create, store and execute procedures and if it were able to determine its subsequent action from decisions, it would be completely logical and general.

Some of the objectives for an extended macro assembly system are the following:-

- The ability to include macro calls in the definitions.
- The use of parenthesis for the nesting of functions or procedures.
- Conditional statements executable at any time.
- Definitions containing facilities to create additional definitions.

All these capabilities, with slight modifications, are included in the TRAC language.

TRAC is a language based on the concept of "macro", as this is used in macro assembly languages. It is designed to be powerful in combining things but its analytical ability is less developed and it might look clumsy used as a scanner and parser.

McIlroy's system was implemented on the IBM 704 computer and it was based on a card format. Like LISP, it used atoms and since it was written in the conventional programmer's format it was rather unsuitable. The problem faced was to modify and generalize it and obtain the results desired from its extended version.

The method needed for generalizing and developing the TRAC language was not evident from the beginning. A complete set of rules was tested tentatively to reach the final goal which was still unclear. If this

set of rules would not allow certain procedures, considered important, then it would be necessary to make changes and alter this set until the desired result had been achieved.

IIA.4 TRAC Syntax

TRAC syntax is simple and very consistent. A TRAC string may contain any number of substrings between its elements. Each substring is enclosed by a pair of parentheses, for example (.....) where the dots represent a string.

There are three different cases, namely #(.....), #(.....), (.....). With the IBM 360 Rax Remote Entry Computing System the "pound" character is preferred rather than the sharp sign. Although referred to in the text as the sharp sign, the character £ will be used in examples from now on. Thus the above examples will be £(.....), ££(.....), (.....).

The expressions £(.....), ££(.....) declare the presence of a TRAC primitive function. The form £(.....) indicates an active function. The form ££(.....) indicates a neutral function. This distinction between the primitive functions is clarified below.

Both active and neutral functions have their interior string subdivided, usually into substrings separated by commas. These substrings are called the arguments of the function.

A pair of parentheses around a string indicates the quotation mode. Any string in quotation mode is protected from functional evaluation.

TRAC strings are evaluated by the TRAC processor which operates according to a scanning algorithm. The scanning process is executed from left to right and the evaluation of nested functional expressions

begins from inside. In the example below the numbers show the order of evaluation:-

```
£(...,£(...,£(...,££(...,££(...),£(...))£(...,££(...),£(...)))))
9     8     7     3     1     2     6     4     5
```

When a function is evaluated, it is replaced by its value, which possibly may be null (a function has null value when after its execution there are no characters left and the entire function disappears). The value of an active function is evaluated further unless it is protected by a pair of parentheses. The value of a neutral function is not evaluated any further.

A string in quotation mode is protected from evaluation during the scanning process. One layer of such protecting parentheses is removed at each scanning, the characters inside are untouched and any included functions are not evaluated.

IIA.5 Primitive Functions of TRAC

The basic instruction format of TRAC is

£(PR-FUN, ARG1, ARG2, ...ARGN) (I)

where PR-FUN stands for primitive function and ARGK stands for the kth argument. TRAC reserves the first argument for the name of a system macro and allows the programmer-defined macros to be specified by another argument. The symbol "£(" has the meaning of the call of a system macro. Thus the argument PR-FUN may be any arbitrary string which must eventually evaluate to one of a fixed set of the system function names.

The macro-name of a TRAC instruction corresponds to the operation field of a general instruction for which the set of allowable operations coincides with the set of primitive TRAC functions.

(I) is the active representation of a function which is the more

often used. The neutral representation is similarly ££(PR-FUN, ...).

The basic set of primitive TRAC functions includes functions that deal with the input and output of strings and string macros that define, call and segment strings. TRAC primitive functions are denoted with a two-letter mnemonic. There are about 35 primitive functions so one character would not be enough.

The main primitive TRAC functions are rs, ps, ds, ss and cl. In the description that follows, their arguments are strings that have been evaluated and they do not contain any active string. The functions are presented in their active form, which is the more common.

IIA.5.1 The read string, £(rs), function

This function has one argument, rs, and as value the string of characters typed from the teletypewriter keyboard until a specified terminating character has been encountered. This character has the interpretation "end of text". It is called the meta character. The apostrophe is used here as such a character. This meta character is deleted from the input string. The language provides facilities for changing the meta character to any other arbitrary character (instead of the apostrophe). The rs function is replaced by its value, which is re-evaluated only in the case of an active function, £(RS).

IIA.5.2 The print string, £(ps,X), function

This function has two arguments, its mnemonic ps and the argument substring X. When there are more than two arguments, the third and subsequent arguments will be ignored and disappear from further computation. For example in £(PS,L1,L2,L10) only the argument L1 would be used.

This function has null value and as effect the actual string X

will be printed on the output medium, usually the teletypewriter. £(PS,X) or ££(PS,X) have the same result which is the output of the string X. This means that for the print string function the result is the same irrespective of whether the function is active or neutral.

IIA.5.3 The define string, £(ds,N,S), function

This function has three arguments. The first is its mnemonic ds, the second the string name and the third the text or content of the string which is stored. This is another null-valued function which results in placing the string S in storage and giving the name N to it. The name N is placed in a table of contents with a pointer to the location of the stored text. A form is a named string which is kept in storage.

If another form with the same name N was defined previously, the new definition destroys the old and the old form is erased from the memory. If the function appears in its neutral form, ££(ds,N,S), the final result is not affected, since this is a null-valued function.

IIA.5.4 The segment string, £(ss,N,S1,S2,...SN), function

This function has three or more arguments. The first is its mnemonic, ss, the second the name of a form in store and the third and any following arguments are the strings S1,S2,... which are examined in turn for matches with the form N. This is another null-valued function.

During the evaluation of this function the form N is obtained from the store. It is first scanned from left to right for a match with the string S1. Whenever a match with the string S1 occurs all matching characters are deleted from the string N and the place is marked to indicate a segment gap of ordinal value 1.

When the matching operation with the string S1 has been carried out the processor scans again from left to right the modified text,

which may now include markers and have some of the original characters deleted, for a match with the next argument, substring S2. Whenever a match is found, a new deletion of the matching characters is made and another marker indicates a segment gap with ordinal value 2. This scanning process continues until the string N has been searched for a match with all the remaining argument substrings of the segment string function. A null string for one of the arguments causes no action for this argument.

This marked string with the segment gap indicators is replaced in memory under the name N. The unchanged parts of the string are called "segments". Such a marked string in store was initially called a form in TRAC terminology but this has been generalized and any text kept in storage is now called a form. The neutral form, ££(ss,N,S1,S2,...SN), has the same final result as the active, since this is another function with null value.

The segment string function creates a macro with dummy variables, the function arguments S1,S2,.... The segment string function can be applied to a string previously segmented. For example the function £(ss,N,X1,X2,...XN) would create new segment gaps with ordinal values 1,2,... and the segment gap indicators would be inserted between the existing ones.

This function causes a rather complex transformation of the string. The final result depends on the order and the number of times in which markers replace occurrences of argument strings in the text.

An important disadvantage in systems like McIlroy's macro assembler and LISP is that this function is limited to acting upon atoms. Each dummy variable S1,S2,... has to be an indivisible entity and as such can match only another one. Matches are not allowed in the middle of the text or in arbitrary character strings. Thus both systems have limited abilities to deal with text in general. They cannot concatenate two

atoms with the removal of commas in between. They cannot divide atoms. In TRAC where there are no atoms the above operations are considered elementary.

IIA.5.5 The call, £(cl,N,X1,X2,...XN), function

This function has two or more arguments. The first is its mnemonic, cl, the second the name of a form in store, and any following are the argument substrings which will replace segment gap indicators existing in the form N. The call function is a function with value. Its value, after being evaluated, replaces the call function. This value is re-evaluated only in case of an active call function.

During the evaluation the form N is brought from storage. It is first scanned from left to right for occurrences of segment gap indicator with ordinal value 1. Wherever such a marker occurs it is deleted and the string X1 is inserted in its place. The TRAC processor continues until all the string has been examined and the string X1 has replaced all occurrences of the ordinal value 1 marker.

When the insertion operation with the string X1 has been carried out the processor scans again for occurrences of segment gap indicator with ordinal value 2, which are replaced with the string X2. This process continues until all the segment gaps of the form have been exhausted.

When the number of argument substrings in the call function is less than the number of segment gap indicators, which are marked by a previous segment string function, the remaining segment gaps will be replaced by null strings. In the opposite case the remaining arguments are ignored.

NSG = number of segment gaps.

NCA = number of arguments in the call function.

1) NSG > NCA N = NSG-NCA, N segment gaps are replaced by null.

2) NSG < NCA N1 = NCA-NSG. The N1 last arguments of the call function are ignored.

As a special case, the function call, £(c1,N), when N is a form with segment gaps, causes the replacement of all of them by null strings. When the form N is not segmented it is brought from storage without any change in it.

The call function results in evaluating its value. This has no side effects on the form N which, if segmented, remains in storage as it was created with the segment string function and, if non-segmented, as it was defined with the define function.

Note (1): The define string function and the segment string function together create a macro-definition where the argument strings S1,S2,...SN represent the dummy variables in the macro-definition. The call string function provides the macro-call and the parameters of this macro-call are the argument substrings X1,X2,...XN of the call function which replace the dummy variables.

Note (2): The define string function can be used to store procedures in the TRAC language. These procedures can be converted to procedural macros by the segment string function. This makes it possible to have macro-calls to procedures with the parameters inserted in the place of dummy variables. Examples showing the above facilities are included in the chapter on TRAC applications (Part IIC below).

IIA.6 The Idling Procedure and Nested Functions

The procedure £(PS,£(RS)), known as the idling procedure, was suggested by Deutsch to control input and output. Until then the input/output of strings and texts had been dealt with in a primitive way, inconsistent with the system. The idea was to introduce two

functions rs, ps for input, output respectively and to use a nested functional expression to control such operations. It has been mentioned already, and it will become clearer later, that nested expressions are evaluated from the inside out.

At the beginning of any processing the idling procedure £(PS,£(RS)) is automatically loaded into the TRAC processor. Then the read string function, rs, is the first to be evaluated. If one types the string

EVERYTHING PASSES, EVERYTHING PERISHES, EVERYTHING PALLS'

this string, up to the apostrophe which is deleted since it is the end read character, is the value of the read string function. This value will take the place of the function rs and it will give

£(PS, EVERYTHING PASSES, EVERYTHING PERISHES, EVERYTHING PALLS)

This print string function will print out the second argument substring and it will give the result

EVERYTHING PASSES, EVERYTHING PERISHES, EVERYTHING PALLS

It should be stressed that whenever a comma is wanted in the text, it must be enclosed between a pair of parentheses otherwise the TRAC processor will be confused and it will take the comma as separator of the argument substrings.[1] The correct way to obtain the previous result is to write the strings in the following way:

EVERYTHING PASSES (,) EVERYTHING PERISHES (,) EVERYTHING PALLS'

Commas are then unquoted by the TRAC processor:

£(PS, EVERYTHING PASSES, EVERYTHING PERISHES, EVERYTHING PALLS)

and the final result is:

EVERYTHING PASSES, EVERYTHING PERISHES, EVERYTHING PALLS

---

[1] The ps function is implemented in a somewhat different way from that described in section IIA.5.2. The differences will be made clear in section IIB.6.

This is the end of the processing cycle which consisted of one rs and one ps function to read a string from the teletypewriter and to print the same string unaltered.

At the completion of any processing, the idling procedure is automatically loaded into the TRAC processor for further processing. If one now types

£(DS, ABC, TEARS(,) IDLE TEARS(,) I KNOW NOT WHAT THEY MEAN)

this will replace the rs function, since it is the value of this function and it will give

£(PS, £(DS, ABC, TEARS, IDLE TEARS, I KNOW NOT WHAT THEY MEAN))

The define string function is executed first. It stores the string TEARS, IDLE TEARS, I KNOW NOT WHAT THEY MEAN under the name ABC. The define function is replaced by its value which is null. So at the end £(PS,) remains and nothing is printed since the second argument substring is null. The processing cycle has finished and the idling procedure is again reloaded into the TRAC processor.

If one now types

£(CL, ABC)

the following steps will take place:

£(PS, £(CL, ABC))

£(PS, TEARS, IDLE TEARS, I KNOW NOT WHAT THEY MEAN)

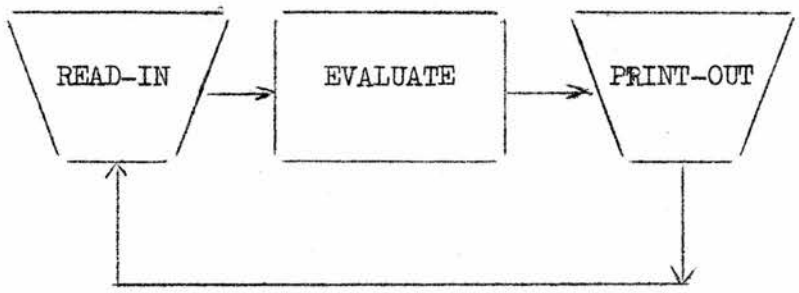
TEARS, IDLE TEARS, I KNOW NOT WHAT THEY MEAN [1]

---

[1] In fact for obtaining the result:
TEARS, IDLE TEARS, I KNOW NOT WHAT THEY MEAN
the define function should be written with double quoted comma as:
£(DS, ABC, TEARS((,)) IDLE TEARS((,)) I KNOW NOT WHAT THEY MEAN)
This is shown in the evaluated examples and it will become clear after describing the TRAC algorithm.

After this suggestion for the idling procedure the philosophy of nested functions was developed and the nested functional expression was fully accepted. All functions have either a null value or, if they have a value, this value replaces the function in the functional expression. Each function can be used as a syntactic marker in some other expression. This syntactic marker would show where the text value of the function has to be inserted in the expression.

The introduced principle of nested functional expressions allows the TRAC processor to be completely general and to treat all expressions uniformly. At the beginning of any computation, the idling procedure is automatically loaded into the area which is reserved as the workspace of the TRAC processor. Any text read from the teletypewriter is the value of the rs function and replaces it. If the text read contains functions, all these functions are evaluated. At the end, any value left from these functions is printed out by the print string function ps and since this is a null function, the workspace of the TRAC processor remains empty finally. At this stage the TRAC processor is re-initialized with the automatic loading of the idling procedure. The forms created previously are not lost but they remain in storage. The TRAC processor is ready now to accept new input for further computation. The idling procedure provides the means for the repetition of the circle



An important conclusion of the above is that any computation involving allowable string processing occurs in the argument substring of the print string function. Since in a functional expression functions can be

nested to any depth, all processing in TRAC occurs within an argument substring of the nest. A very essential feature of the language is that argument substrings of a function can themselves be functions with other argument substrings and so on. Thus it is possible to have any depth in nesting functions and to extend the capabilities of the language.

IIA.7 Examples on the Main Primitive Functions

Some examples are given (pp. 49-54) to show how the main primitive functions operate on strings. These examples are followed by their evaluation with the use of the idling procedure which controls input/output and the processing in general and is reloaded at each new processing cycle.

IIA.7.1 The read string function

The **rs** function wherever it occurs makes it possible for any string to be read as input from the teletypewriter. Since the apostrophe is the "end read" character, anything written after it is not read but is considered as comment.

In Ex-1, RS, the string typed is the value of the rs function and replaces this function in the idling procedure. There it becomes the second argument substring of the ps function and as such it is printed out unaltered.

Similar is the EX-2, RS.

IIA.7.2 The print string function

Wherever such a function occurs, irrespective of its place in any functional expression, it will print out the second argument substring.

(contd. p. 55)

```
£(TN)'
EVERYTHING PASSES,EVERYTHING PERISHES,EVERYTHING PALLS'
EVERYTHING PASSES(,) EVERYTHING PERISHES(,) EVERYTHING PALLS'
£(DS,ABC,TEARS(,)IDLE TEARS(,)I KNOW NOT WHAT THEY MEAN)'
£(CL,ABC)'
£(DS,ABC,TEARS((,))IDLE TEARS((,))I KNOW NOT WHAT THEY MEAN)'
£(CL,ABC)'
GIVE ME BUT ONE FIRM SPOT ON WHICH TO STAND AND I WILL MOVE THE EARTH'EX-1,RS
THE LIVING NEED MORE CHARITY THAN THE DEAD'EX-2,RS
£(PS,PLATO IS DEAR TO ME BUT DEARER STILL IS TRUTH)'EX-1,PS
££(PS,AND WE FORGET BECAUSE WE MUST NOT BECAUSE WE WILL)'EX-2,PS
£(DS,XX,ABSENCE MAKES THE HEART GROW FONDER)'EX-1,DS
££(DS,ABC,£(RS))'EX-2,DS
IF IT IS NOT TRUE IT IS A HAPPY INVENTION'
£(DS,£(RS),TIMES CHANGE AND WE CHANGE WITH THEM)'EX-3,DS
HARRISSON'
££(DS,ARISTOTLE,MAN IS BY NATURE A POLITICAL ANIMAL)'EX-4,DS
£(DS,FF,THE SUN IS BRIGHT)'EX-5,DS
£(CL,XX)'EX-1,CL
£(CL,ABC)'EX-2,CL
£(CL,HARRISSON)'EX-3,CL
£(CL,ARISTOTLE)'EX-4,CL
£(CL,FF)' EX-5,CL
£(SS,ARISTOTLE,NATURE,POLITICAL)'EX-4,SS
££(SS,FF,SUN,BRIGHT)'EX-5,SS
£(CL,ARISTOTLE)'EX-41,CL
£(CL,ARISTOTLE,MARKER1,MARKER2)'EX-42,CL
£(CL,ARISTOTLE,NATURE)'EX-43,CL
£(CL,ARISTOTLE,NATURE,POLITICAL)'EX-44,CL
£(CL,ARISTOTLE,NATURE,POLITICAL,OTHER)'EX-45,CL
£(CL,FF)'EX-51,CL
£(CL,FF,GRASS,GREEN)'EX-52,CL
£(CL,FF,MOON,YELLOW)'EX-53,CL
£(CL,FF,SUN,BRIGHT)'EX-54,CL
'
```

```
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++
PS"
+++++THIS PS FUNCTICN HAS NULL STRING+++++
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++
RS
PS"EVERYTHING PASSES"EVERYTHING PERISHES"EVERYTHING PALLS
EVERYTHING PASSES"EVERYTHING PERISHES"EVERYTHING PALLS
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++
RS
PS"EVERYTHING PASSES, EVERYTHING PERISHES, EVERYTHING PALLS
EVERYTHING PASSES, EVERYTHING PERISHES, EVERYTHING PALLS
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++
RS
DS"ABC"TEARS,IDLE TEARS,I KNCW NCT WHAT THEY MEAN
PS"
+++++THIS PS FUNCTICN HAS NULL STRING+++++
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++
RS
CL"ABC
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++
+++++FCRM WITH NO MARKERS ,ARGUMENTS CF CL FUNCTION IGNORED+++++
PS"TEARS"IDLE TEARS"I KNOW NCT WHAT THEY MEAN
TEARS"IDLE TEARS"I KNOW NCT WHAT THEY MEAN
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++
RS
DS"ABC"TEARS(,)IDLE TEARS(,)I KNCW NCT WHAT THEY MEAN
PS"
+++++THIS PS FUNCTICN HAS NULL STRING+++++
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++
RS
CL"ABC
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++
+++++FCRM WITH NO MARKERS ,ARGUMENTS CF CL FUNCTION IGNORED+++++
PS"TEARS,IDLE TEARS,I KNOW NCT WHAT THEY MEAN
TEARS,IDLE TEARS,I KNOW NCT WHAT THEY MEAN
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++
RS
PS"GIVE ME BUT CNE FIRM SPCT CN WHICH TC STAND AND I WILL MOVE THE EARTH
```

```
GIVE ME BUT ONE FIRM SPOT ON WHICH TO STAND AND I WILL MOVE THE EARTH
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++
RS
PS"THE LIVING NEED MORE CHARITY THAN THE DEAD
THE LIVING NEED MORE CHARITY THAN THE DEAD
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++
RS
PS"PLATO IS DEAR TO ME BUT DEARER STILL IS TRUTH
PLATO IS DEAR TO ME BUT DEARER STILL IS TRUTH
PS"
+++++THIS PS FUNCTION HAS NULL STRING+++++
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++
RS
PS"AND WE FORGET BECAUSE WE MUST NOT BECAUSE WE WILL
AND WE FORGET BECAUSE WE MUST NOT BECAUSE WE WILL
PS"
+++++THIS PS FUNCTION HAS NULL STRING+++++
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++
RS
DS"XX"ABSENCE MAKES THE HEART GROW FONDER
PS"
+++++THIS PS FUNCTION HAS NULL STRING+++++
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++
RS
RS
DS"ABC"IF IT IS NOT TRUE IT IS A HAPPY INVENTION
PS"
+++++THIS PS FUNCTION HAS NULL STRING+++++
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++
RS
RS
DS"HARRISSON"TIMES CHANGE AND WE CHANGE WITH THEM
PS"
+++++THIS PS FUNCTION HAS NULL STRING+++++
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++
RS
DS"ARISTOTLE"MAN IS BY NATURE A POLITICAL ANIMAL
PS"
```

```
+++++THIS PS FUNCTION HAS NULL STRING+++++                                77
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                             78
RS                                                                       79
DS"FF"THE SUN IS BRIGHT                                                  80
PS"                                                                      81
+++++THIS PS FUNCTION HAS NULL STRING+++++                                82
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                             83
RS                                                                       84
CL"XX                                                                    85
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                         86
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++          87
PS"ABSENCE MAKES THE HEART GROW FONDER                                   88
ABSENCE MAKES THE HEART GROW FONDER                                      89
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                            90
RS                                                                       91
CL"ABC                                                                   92
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                        93
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++         94
PS"IF IT IS NOT TRUE IT IS A HAPPY INVENTION                             95
IF IT IS NOT TRUE IT IS A HAPPY INVENTION                                96
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                            97
RS                                                                       98
CL"HARRISSON                                                             99
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                       100
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++        101
PS"TIMES CHANGE AND WE CHANGE WITH THEM                                102
TIMES CHANGE AND WE CHANGE WITH THEM                                   103
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                          104
RS                                                                     105
CL"ARISTOTLE                                                           106
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                      107
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++       108
PS"MAN IS BY NATURE A POLITICAL ANIMAL                                 109
MAN IS BY NATURE A POLITICAL ANIMAL                                    110
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                          111
RS                                                                     112
CL"FF                                                                  113
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                      114
```

```
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
PS"THE SUN IS BRIGHT
THE SUN IS BRIGHT
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++
RS
SS"ARISTOTLE"NATURE"POLITICAL
PS"
+++++THIS PS FUNCTION HAS NULL STRING+++++
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++
RS
SS"FF"SUN"BRIGHT
PS"
+++++THIS PS FUNCTION HAS NULL STRING+++++
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++
RS
CL"ARISTOTLE
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++
PS"MAN IS BY  A  ANIMAL
MAN IS BY  A  ANIMAL
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++
RS
CL"ARISTOTLE"MARKER1"MARKER2
PS"MAN IS BY MARKER1 A MARKER2 ANIMAL
MAN IS BY MARKER1 A MARKER2 ANIMAL
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++
RS
CL"ARISTOTLE"NATURE
+++++NUMBER OF ARGUMENTS LESS THAN NO OF MARKERS
NOMARKERS-NOARG MARKERS REPLACED BY NULL+++++
PS"MAN IS BY NATURE A  ANIMAL
MAN IS BY NATURE A  ANIMAL
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++
RS
CL"ARISTOTLE"NATURE"POLITICAL
PS"MAN IS BY NATURE A POLITICAL ANIMAL
MAN IS BY NATURE A POLITICAL ANIMAL
```

```
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                153
RS                                                                           154
CL"ARISTOTLE"NATURE"POLITICAL"OTHER                                          155
PS"MAN IS BY NATURE A POLITICAL ANIMAL                                       156
MAN IS BY NATURE A POLITICAL ANIMAL                                          157
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                158
RS                                                                           159
CL"FF                                                                        160
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                            161
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++              162
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++              163
PS"THE  IS                                                                   164
THE  IS                                                                      165
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                166
RS                                                                           167
CL"FF"GRASS"GREEN                                                            168
PS"THE GRASS IS GREEN                                                        169
THE GRASS IS GREEN                                                           170
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                171
RS                                                                           172
CL"FF"MOON"YELLOW                                                            173
PS"THE MOON IS YELLOW                                                        174
THE MOON IS YELLOW                                                           175
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                176
RS                                                                           177
CL"FF"SUN"BRIGHT                                                             178
PS"THE SUN IS BRIGHT                                                         179
THE SUN IS BRIGHT                                                            180
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                181
RS                                                                           182
+++++TRAC JOB FINISHED THANK YOU GOOD-BYE,YEIA-XARA COSTAS+++++              183
```

In the EX-1, PS, the ps function is active and in the EX-2 it is neutral. This makes no difference to the final result since the function has null value and nothing remains after the printing of its argument substring.

The evaluation of the EX-1, PS, begins with the loading of the idling procedure.

The rs function is evaluated first and is replaced by its value which is typed from the teletypewriter. This is the print string function.

This print string function is now evaluated and prints out its second argument substring.

The value of the function is null so nothing remains in its place. The idling procedure looks like £(PS,) and it prints out nothing since the second argument is null.

The processing cycle finishes and the TRAC processor is re-initialized by reloading the idling procedure.

Similarly in the example EX-2, PS.

The other example of the ps function is evaluated in a similar way.

IIA.7.3 The define string function

Any string can be named and stored by using this function. EX-1, EX-2, EX-3, EX-4, EX-5 DS show how this function is evaluated. EX-2 and EX-5 DS include the function in its neutral form. Since ds is a null value function this does not change the action taken which is the naming and storing of some string.

The evaluation of the EX-2, DS, is described in detail. The idling procedure, already loaded, causes the rs function to be evaluated first. The rs function is replaced by its value which is typed from the teletypewriter. This value is the string ££(DS, ABC, £(RS))'.

The functional expression after the substitution looks like

£(PS, ££(DS, ABC, £(RS)))'

3 2 1

where the numbers show the order of evaluation.

The rs function is replaced again by its value which is the string:

IF IT IS NOT TRUE IT IS A HAPPY INVENTION'

The expression then takes the form:

£(PS, ££(DS, ABC, IF IT IS NOT TRUE IT IS A HAPPY INVENTION))'

The define string function is evaluated and results in recording the string IF IT IS NOT TRUE IT IS A HAPPY INVENTION in storage under the name ABC. After that it is replaced by its null value and the expression takes the form £(PS,)'.

The print function prints out nothing since the second argument is null. The processing cycle finishes and the idling procedure is loaded again. In a similar way the other examples of the ds function are evaluated.

All these defined strings can be brought from storage by means of a simple call function with two arguments, the second being the string's name. EX-1, EX-2, EX-3, EX-4, EX-5, CL demonstrate this possibility.

The evaluation of the EX-3, CL is described in detail. The steps during the evaluation are mentioned in the order in which they occur. The explanation is similar to EX-2, DS:

£(PS, £(RS))'	the idling procedure
£(RS)'	the rs function is evaluated first
£(CL, HARRISSON)'	this is the string read
£(PS, £(CL, HARRISSON))'	rs function replaced by its value
£(PS, TIMES CHANGE AND WE CHANGE WITH THEM)'	

The value of the cl function replaces it. This value is the string stored under the name HARRISSON.

TIMES CHANGE AND WE CHANGE WITH THEM, the ps function is evaluated

and prints out the second argument substring. The workspace of the processor is cleared since the ps function has null value and the idling procedure is loaded again.

In a similar way are evaluated the other examples referring to this function.

IIA.7.4 The segment string function

This function can introduce dummy or formal variables into a string. This is the way to create the text for a macro. The function has null value and the action taken does not change even when it is presented in its neutral form. EX-4, EX-5, SS show the way this function is evaluated. One of them, EX-5, SS, is described fully; the other is evaluated in a similar way.

The evaluation of EX-5, SS is described step by step in order of occurrence:

£(PS, £(RS))'	the idling procedure
£(RS)'	the rs function is evaluated first
££(SS, FF, SUN, BRIGHT)'	this is the string read
£(PS, ££(SS, FF, SUN, BRIGHT))	the rs function is replaced by its value

The segment string function is evaluated. Markers are created for the segment gaps and the segmented form is replaced in memory under the same name. After evaluation it is replaced by its null value and the expression takes the form £(PS,).

This prints out nothing. It is replaced by its null value and the idling procedure is loaded again, since this is the end of another processing cycle.

In this way, two dummy variables have been introduced in the string FF.

IIA.7.5 The call function

This function provides the means for a macro call. Parameters are now inserted in the place of dummy variables which existed as the result of a segment string function.

EX-41 to EX-45, CL and EX-51 to EX-54, CL show some of the possible uses of such a function. In the following paragraphs all examples are examined to justify the results they produce.

EX-41, CL results in MAN IS BY A ANIMAL. The form ARISTOTLE is segmented with two markers. This function call provides no arguments and the markers of the form are replaced by null.

EX-42, CL results in MAN IS BY MARKER1 A MARKER2 ANIMAL, the arguments of the cl function, MARKER1 and MARKER2, replacing the corresponding segment gap indicators.

EX-43, CL results in MAN IS BY NATURE A ANIMAL. The number of arguments in the cl function is less than the number of existing segment gap indicators. In this example there is one marker in the form ARISTOTLE without corresponding argument in the cl function. This marker is replaced by null.

EX-44, CL results in MAN IS BY NATURE A POLITICAL ANIMAL. Here the number of segment gap markers of the form ARISTOTLE and the number of arguments in the call function are equal.

EX-45, CL gives the same result as the previous example. Here the number of arguments in the cl function is greater than the number of segment gap markers. The remaining arguments, in this case the argument OTHER, are ignored.

EX-51, CL results in THE IS, the reason being exactly the same as in EX-41, CL.

EX-52, EX-53 and EX-54, CL demonstrate the use of the cl function as a macro call. By changing each time the parameters which replace

the dummy variables it is possible to change the original text to, respectively:

THE GRASS IS GREEN

THE MOON IS YELLOW

THE SUN IS BRIGHT

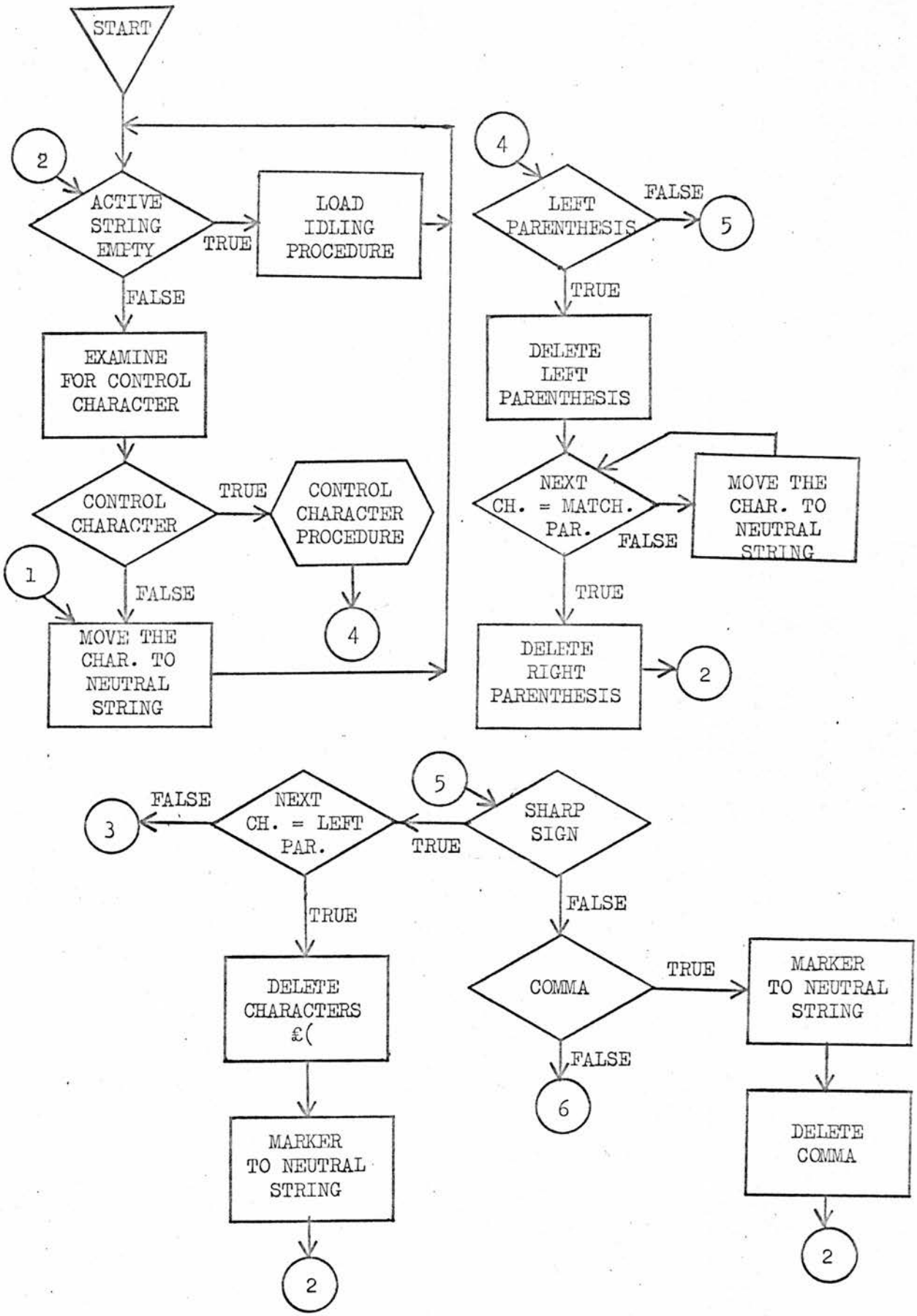
IIA.8 The TRAC Algorithm

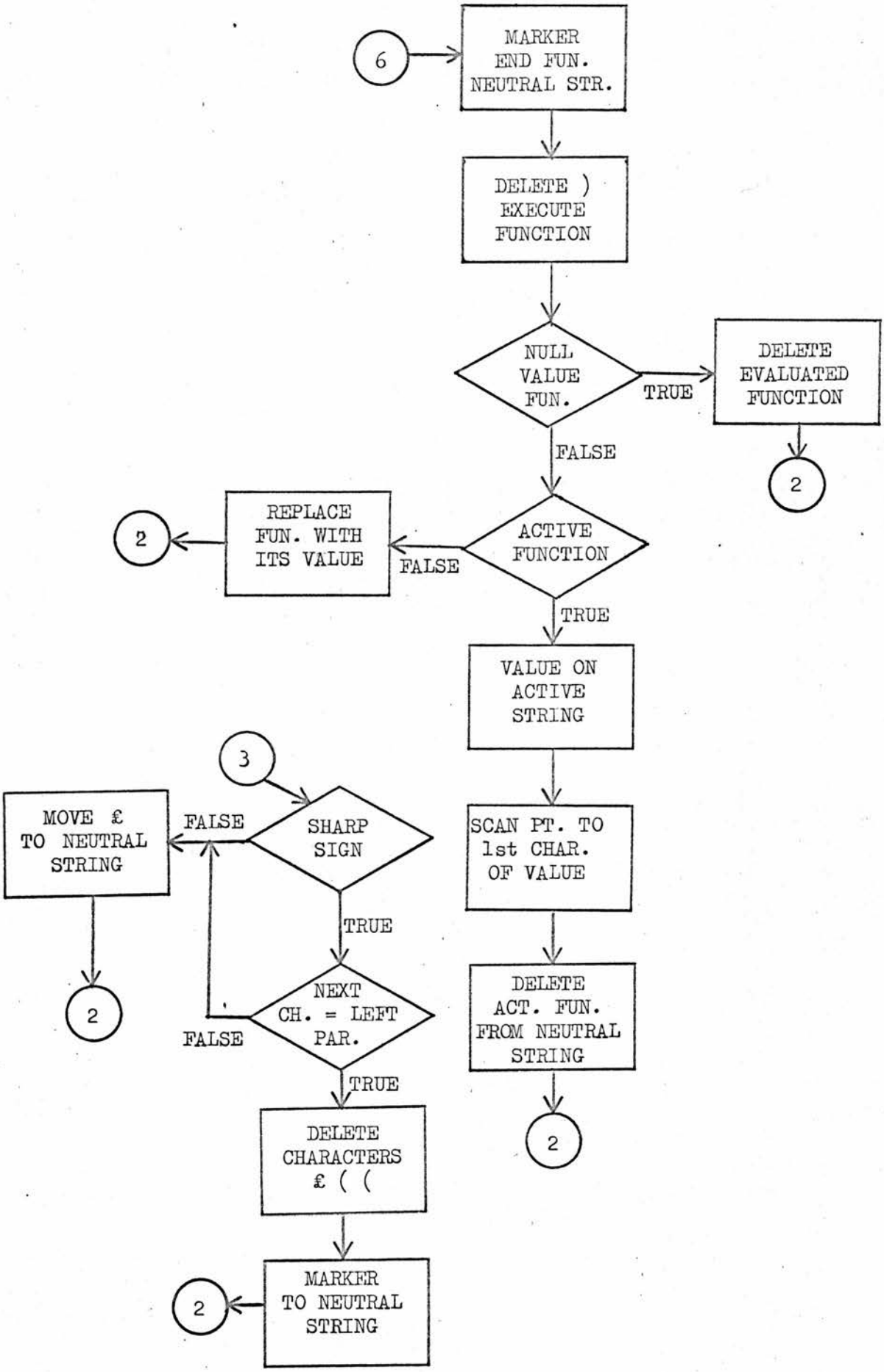
A scanning algorithm determines the way any function or functional expression is evaluated in TRAC. The algorithm operates on any nested expression from left to right and from outside in. It starts with the print string function of the idling procedure. Each function is evaluated when all its argument substrings have been evaluated. They are converted from the active mode, where they might contain unevaluated functions, to the neutral mode. An expression is said to be in the neutral mode when all operations allowed by the TRAC scanning algorithm have been performed on it. In a nested functional expression the inner function which contains no arguments in the active mode is evaluated first.

At the beginning of the processing, all unevaluated expressions are on the active string with the scanning pointer to the leftmost character of the string. After evaluation each character is transferred to the right-hand end of the neutral string. It contains only evaluated characters and for this reason it is called neutral. A pointer, the current location pointer, indicates the next available position in the neutral string.

A flowchart of the scanning algorithm and some explanatory notes that follow will demonstrate the principles and operations of the algorithm.

FIA.9 Flowchart of TRAC Algorithm




6
MARKER END FUN. NEUTRAL STR.
DELETE) EXECUTE FUNCTION
NULL VALUE FUN.
TRUE
DELETE EVALUATED FUNCTION
2
FALSE
ACTIVE FUNCTION
FALSE
REPLACE FUN. WITH ITS VALUE
2
TRUE
VALUE ON ACTIVE STRING
SCAN PT. TO 1st CHAR. OF VALUE
DELETE ACT. FUN. FROM NEUTRAL STRING
2
3
SHARP SIGN
FALSE
MOVE £ TO NEUTRAL STRING
2
TRUE
NEXT CH. = LEFT PAR.
FALSE
TRUE
DELETE CHARACTERS £ ((
MARKER TO NEUTRAL STRING
2

IIA.10 Description of the Algorithm

A. The character pointed to by the scanning pointer is examined. When there is no character left, which means all expressions in the active string have been evaluated, the idling procedure is reloaded and evaluation begins with the print string function.

B. If the character pointed to is not a control character,[1] it is transferred to the neutral string.

B1. The scanning pointer is set to the next character of the active string. Then control is transferred to A.

C. 1/. If the control character is a left parenthesis all characters following it, until the occurrence of a matching right parenthesis, are moved to the neutral string unaltered. The pair of enclosing parentheses is deleted. Then control is transferred to B1.

2/. If the control character is a sharp sign, the next character is examined.

2.1/. If the next character is a left parenthesis, an active function occurs. The characters £ (are deleted. The current location of the neutral string is marked to show the beginning of an active function and a new argument substring. Then control is transferred to B1.

2.2/. If the next character is another sharp sign, the following character is examined.

2.21/. If it is a left parenthesis a neutral function occurs. The characters £ £ (are deleted. The current location of the neutral string is marked to show the beginning of a neutral function and a new argument substring. Then control is transferred to B1.

---

[1] Control characters in TRAC are £ , ( ) namely, sharp sign, comma, left parenthesis, right parenthesis.

2.22/. If it is not a left parenthesis the first £ is moved to the neutral string. Then control is transferred to B1.

2.3/. Go to 2.22.

3/. If the control character is a comma, it is deleted and the current location of the neutral string is marked to show the end of a substring and the beginning of a new one.

4/. If the control character is a right parenthesis, this is the end of a function. The current location of the neutral string is marked to show the end of a function, the parenthesis is deleted, and the appropriate action for the function is taken.

4.1/. If the function just ended has null value, after being evaluated it is deleted from the neutral string. Then control is transferred to B1.

4.2/. If the function has value:

4.21/. If the function is active, its value is inserted in the active string in front of the first unevaluated character. This is accomplished by moving the unevaluated characters of the active string left or right depending on the number of characters of the value of the active function. The scanning pointer of the active string is set to the first character of this value. The evaluated active function is deleted from the neutral string. Then control is transferred to B1.

4.22/. If the function is neutral it is replaced by its value in the neutral string. Then control is transferred to B1.

Note: The characters for "carriage return", "line feed", and "tabulate" if available to the keyboard are deleted by the scanning algorithm unless they are enclosed by a pair of parentheses and thus are in the quote mode. This facility improves readibility since it allows the

user when a procedure is complicated to write the functions in a column rather than in a line across the page.

In the following example, an attempt is made to clarify the scanning processof the TRAC algorithm.

Some symbols are introduced to represent pointers of the neutral string:

a, active function pointer

n, neutral function pointer

c, pointer for comma

e, end of function pointer.

Consider the Ex-2, DS in Section IIA.7.3 above. The evaluation by the TRAC scanning algorithm presents various phases which are shown below in the order in which they occur in the active and neutral strings.[1]

ACTIVE string	NEUTRAL string
Empty	Empty
£(PS,£(RS)) 1	aPScaRSe$_1$
££(DS,ABC,£(RS))) 2 3	aPScn$_2$DScABCcaRSe$_3$
IF IT IS NOT TRUE IT IS A	apscn$_2$DScABCcIF IT IS NOT TRUE
HAPPY INVENTION)) 45	IT IS A HAPPY INVENTION$_4$
	aPSce$_5$
	Empty

[1] Numbers show correspondence between active and neutral strings.

IIA.11 Modes of Evaluation in TRAC

There are three modes in evaluating strings in TRAC:-

(i) The quote mode (- - - - - -)

(ii) The active mode £(- - - - - -)

(iii) The neutral mode ££(- - - - - -).

When a string is in the quote mode it is protected from evaluation. The pair of enclosing parentheses is deleted by the scanning algorithm and the characters in between are passed to the neutral string without evaluation of any kind.

When a functional expression is evaluated either it has a value or its value is the null string. A null-valued function, whether it is in the active or neutral form, is deleted from the neutral string.

If a function has a value and the value string was produced by an active function, it has to be scanned. If it contains other functions, they are evaluated and so on until there is no active function left to produce any value. If the value string were produced by a neutral function, the value is treated as an evaluated string and it is put on the neutral string, although it may contain unevaluated functions.

Examples follow to demonstrate the three modes of evaluation. Suppose the following TRAC statements are typed in the following order 1.

1. £(DS, XXX, TRUTH SITS UPON THE LIPS OF DYING MEN)'
2. £(DS, YYY, (£(CL, XXX)))'
3. £(PS, (£(CL, YYY)))'
4. £(PS, ££(CL, YYY))'
5. £(PS, £(CL, YYY))'

The 3rd statement shows the quote mode and the result is £(CL, YYY).

The 4th " " " neutral " " " " " £(Cl, XXX).

The 5th " " " active " " " " " "

TRUTH SITS UPON THE LIPS OF DYING MEN.

The processing of these functions by the scanning algorithm becomes evident by showing the successive steps of evaluation on the active and neutral strings.

ACTIVE string	NEUTRAL string
St.1/ Empty	Empty
£(PS,£(RS)) 1	aPScaRSe$_1$
£(DS,XXX,TRUTH SITS UPON THE	aPScaDScXXXcTRUTH SITS UPON
LIPS OF DYING MEN)) 23	THE LIPS OF DYING MENe$_2$
	aPSce$_3$
	Empty
St.2/ Empty	
£(PS,£(RS)) 1	aPScaRSe$_1$
£(DS,YYY,(£(CL,XXX)))) 234	aPScaDScYYYc£(CL,XXX)e$_3$
	aPSce$_4$
	Empty
St.3/ Empty	
£(PS,£(RS)) 1	aPScaRSe$_1$
£(PS,(£(CL,YYY)))) 23	aPScaPScYYYc£(CL,YYY)e$_2$
	aPSce$_3$
	Empty
St.4/ Empty	
£(PS,£(RS)) 1	aPScaRSe$_1$
£(PS,££(CL,YYY))) 234	aPScaPScnCLcYYYe$_2$
	aPScaPSc£(CL,XXX)e$_3$
	aPSce$_4$
	Empty

St.5/ Empty	
£(PS,£(RS)) 1	aPScaRSe$_1$
£(PS,£(CL,YYY))) 2	aPScaPScaCLcYYYe$_2$
£(CL,XXX))) 3	aPScaPScaCLcXXXe$_3$
TRUTH SITS UPON THE LIPS OF	aPScaPScTRUTH SITS UPON THE
DYING MEN)) 45	LIPS OF DYING MENe$_4$
	aPSce$_5$

IIA.12 The TRAC Primitive Functions

The set of the chosen primitive functions is subdivided into subsets in two ways:-

(i) According to their value. There are two subsets:-

(a) Null-valued functions. This set contains the following functions:

£(cm, X),[1] £(ps, X), £(ds, N, S), £(ss, N, S1, Sq, ...SN)

£(cr, N), £(dd, N1, N2, ...Nn), £(da), £(sb, N, N1, N2, ...Nn)

£(fb, N), £(eb, N), £(ln, X), £(pf, N), £(tn), £(tf), £(sd, X)

(b) Functions with value:

£(rs), £(rc), £(cl, N, X1, X2, ...Xn), £(cs, N, Z), £(cc, N, Z)

£(cn, N, D, Z), £(in, N, X, Z), £(ad, D1, D2, Z)

£(su, D1, D2, Z), £(ml, D1, D2, Z), £(dv, D1, D2, Z),

£(bu, O1, O2), £(bi, O1, O2), £(bc, O1), £(bs, D1, O1)

£(br, D1, O1), £(eq, X1, X2, X3, X4), £(gr, D1, Dq, X1, X2)

(ii) According to the operations they perform. There are

[1] Here the first argument, the mnemonic of the function, is written with small letters to distinguish it from the other argument sub-strings. Elsewhere these mnemonics are written with capital letters.

seven subsets:

(a) Input-output:

£(rs), £(rc), £(cm, X), £(ps, X), £(sd, X)

(b) Define and call functions:

£(ds, N, X), £(ss, N, S1, S2, ...Sn), £(cl, N, X1, X2, ...Xn),
£(cs, N, Z)

£(cc, N, Z), £(cn, N, D, Z), £(in, N, X, Z), £(cr, N)

£(dd, N1, N2, ...Nn), £(da)

(c) Arithmetic functions:

£(ad, D1, D2, Z), £(su, D1, D2, Z), £(ml, D1, D2, Z), £(dv, D1, D2, Z)

(d) Boolean functions:

£(bu, O1, O2), £(bi, O1, O2), £(bc, O1), £(bs, D1, O1) £(br, D1, O1)

(e) Decision functions:

£(eq, X1, X2, X3, X4), £(gr, D1, D2, X1)

(f) External storage management:

£(sb, N, N1, N2, ...Nn), £(fb, N), £(eb, N)

(g) Diagnostic functions:

£(ln, N), £(pf, N), £(tn), £(tf)

One of the essential characteristics of the TRAC language is that it is not dependent on the chosen set of functions. If another set of functions would be more convenient for performing a task, the original set could be changed easily to include new functions and to delete existing ones or even to become a completely new set including functions performing different operations. The main contribution of TRAC to the development of macro processing is its scanning algorithm, the facility for nesting functions by allowing the argument substrings of a function to contain other functions, and the various modes of evaluation, namely the quote mode, the active mode and the neutral mode.

The five main primitive functions rs, ps, ds, ss, cl have been described already. The remainder are shown in their active form, in which they appear more often. As the functions appear below they do not contain unevaluated functions in their arguments.

1/. The £(rc), read character, function is similar to the read string function. Its value contains the next character as it is received from the teletypewriter. It can be any character available on the keyboard, even the end of the rs function meta character.

2/. The £(cm, X), change meta, function has null value. It allows the user to change the meta character with the first character of the argument substring X. When the processing starts the TRAC processor is loaded with a standard meta character, usually the apostrophe.

3/. The £(sd, X), select device, function provides control of peripheral units. It is a function with null value and X is the mnemonic of the device selected. This function can modify the following rs, rc, or ps function to operate on the device named. For example £(sd, p) £(rs) uses the paper tape reader for input.

Each form stored, in TRAC memory, has associated with it a form pointer. Each time a certain number of characters of the form are retrieved from store, the form pointer is increased accordingly. The following four functions allow operations to be performed on parts of strings.

4/. The £(cs, N, Z), call segment, function contains three arguments. The value is a string from the current position of the form N pointer to its right until the first segment gap indicator. When the form N is null or when the form pointer points after the last character the value of the function is the string Z. After execution of the function, the form pointer is set to the first character after the segment gap.

5/. The £(cc, N, Z), call character, function contains three

arguments. The value is the character of the form N pointed to by its pointer. When the form N is null or when the pointer points after the last character, the value of the function is the string Z. After execution the pointer moves one character.

6/. The £(cn, N, D, Z), call n characters, function contains four arguments. Its value is some string of characters obtained from the form N. The number of characters for the value string is specified by the decimal integer number contained at the tail end of the substring D. The first character of the value is the one pointed to by the pointer. The remaining characters of the value are taken either from the string following this character if the decimal number is positive or from the string preceding this character if the number is negative. The form pointer is increased or decreased to point at the first character after or before the last character of the value respectively. If there is not a sufficient number of characters the value is the string Z.

7/. The £(in, N, X, Z), initial, is another function with four arguments. The form N is searched for on occurrence of the string X. Starting character is the one pointed to by the form pointer. If such a string occurs, the value is the string starting from the pointer up to and including the last character before the first matching character. The pointer is increased to point to the first character after the matching string. If there is no match the value is the substring Z and the form pointer remains unchanged.

8/. The £(cr, N), call restore, function has null value and two arguments. It resets the pointer of the form N to point at its initial character.

9/. The £(dd, N1, N2, ...Nn), delete definition, function can have an indefinite number of arguments. It is a null-valued function. It deletes the forms named N1, N2, ...Nn from store and it removes

their names from the list of names.

10/. The £(da), delete all, function has null value and one argument. It deletes all existing forms from storage and it erases their names.

11/. The arithmetic functions. TRAC as a string manipulation language has limited capabilities in numerical computation. It provides facilities for integer arithmetic to deal with arrays. There are functions to perform addition, subtraction, multiplication and division with integers, having mnemonics respectively ad, su, ml, dv. They accept decimal numbers as arguments and the value produced is decimal. They are stored in decimal representation in the TRAC memory but the arithmetic is done in binary. The decimal numeric digits are found at the tail ends of the argument substrings. The string preceding the first argument is reserved and it precedes the decimal answer but the one that precedes the second argument is ignored. Negative quantities are indicated by a "minus sign", -, and leading zeros are ignored. If the result value causes an overflow the value of the arithmetic function is the string Z which is treated as an active function. The £(ad, D1, D2, Z) function is equivalent to D1 = D1 + D2 and similarly:

£(su, D1, D2, Z) D1 = D1 - D2

£(ml, D1, D2, Z) D1 = D1 - D2

£(dv, D1, D2, Z) D1 = largest integer $\leq$ D1/D2

where the "=" has the meaning at replacement and D1, D2 represent the decimal values of the arguments.

12/. Boolean functions. TRAC provides facilities for operation on strings of bits of value 0 or 1 **i.e.** Boolean vectors. The strings are represented by a sequence of octal digits. The existing functions are Boolean union, intersection, complement, shift and rotate:

that is, £(bu, S1, S2) £(bi, S1, S2), £(bc, S1) £(bs, D1, S1), £(br, D1, S1). The bit strings are right justified. In union the shorter string is filled with leading zeros, in intersection it is truncated at the left. In the complement, shift and rotate the length of the string does not change. Shift and rotate are either to the left — D1 places when D1 $>$ 0 — or to the right — D1 places when D1 $<$ 0.

13/. Decision functions. The TRAC language provides facilities for branching. The branch on equality £(eq, X1, X2, X3, X4) function is based on the equality of two strings. If the string X1 is identical to the string X2, the value is the string X3, if not the value is the string X4. The strings X3, X4 as argument substrings may contain text or functions or even procedures. Thus with this function it is possible to transfer control to another procedure. The function £(gr, X1, X2, X3, X4) allows the comparison of numerical quantities. The value is X3 or X4 according to whether X1 is greater or equal to X2 or not. X1, X2 are decimal values at the tail ends of the second and third arguments.

14/. External storage management functions. TRAC allows the use of back-up store and provides facilities to store forms and to retrieve them. The null-valued £(sb, N, N1, N2, ...Nn), store block, function causes all forms N1, N2, ...Nn etc. to be stored as one block in a mass storage device, disc or drum with all form pointers and segment gaps saved. When the forms are moved to the external storage they are erased from storage and a new form N is created which contains the address of the block in the external storage.

The £(fb, N), "fetch block" performs the opposite operation. It retrieves the forms from the external storage without erasing them. The name N is the name of the block to be fetched. The form N is not erased from the main store. The retrieved forms remain unchanged and they still contain their names, their pointers and the segment gap

indicator.

The £(eb, N) "erase block". This null-valued function erases the form N from the main storage and the corresponding block with all forms from the external storage.

The above functions allow forms to be stored in external media. Thus the forms are protected from accidental erasure which might happen in the main store and the capabilities of the system are expanded, since more store can be used. A tree data structure can be built where a group of forms is stored under a group name, a set of group names under a section name and so on.

15/. Diagnostic functions. These are important for dealing with complex procedures. The £(ln, X), list names, is a null-valued function which prints out the names of all the forms stored in TRAC memory. Each printed name is preceded by the string X. The £(pf, N), print form, is a null-valued function with two arguments. It prints out the form named N, including some indication for each segment gap and its ordinal value. It makes it possible to have a complete picture of the form in storage.

The £(tn), trace on, is another null-valued function. When it is applied, it provides a step-by-step trace of executed TRAC procedures. Whenever a function is built up on the neutral string it is typed out to be inspected. The trace function causes no further action and the process continues normally, until another function is again typed out from the neutral string.

The £(tf), trace off, is a null-valued function which halts the trace mode.

The following example shows the capabilities of the TRAC language in defining procedures. The factorial of a number is calculated by using simple recursion:

```
£(ds, factorial, (£(eq, 1, X, 1,
(£(ml, X, £(cl, factorial, £(ad, X, -1))))
)))£(ss, factorial, X)'
```

The call £(cl, factorial, 5)' produces the result 120. More examples for the TRAC functions and the use of procedures are included in the applications.

IIA.13 Main Features of the TRAC Language

The main features of the language are

1. The literal string: In TRAC any string of characters represents itself. A string is viewed as an object at the literal level. This is unusual among other symbol manipulation languages where operations on strings are performed usually at the name level. TRAC was designed this way in order to overcome what was believed to be a drawback in string processing, namely the "name level" of strings.

The language provides also facilities for considering a string as a name for some other string. For example, in the function £(CL, XXYY, A, B, C) the string XXYY is considered to be the name of another stored string. However, a string name is not automatically replaced by its value. There is a TRAC function, for example CL, SS, which determines the way of interpreting its arguments. Furthermore if there is a value the distinction of the function in either £(, active, or £&(, neutral, determines subsequent action on this value.

In TRAC when a string is defined as a name, further information specifies how this string is to be treated. There is no automatic replacement of strings with their values. Since TRAC is designed to operate on literary text, it would be unreasonable and very complex to generate names for all argument substrings of its functions. The

facility provided to operate on the literal string makes the language powerful and flexible.

2. The use of functional syntax: The treatment of functions in TRAC either as £(-----), active, or ££(-----), neutral, is unique among macro-processing languages. As already stated, the difference is in their value, which is evaluated when the function is active and not when the function is neutral. The evaluation continues until there is no active function produced as a value. A single pair of parentheses defines the quote mode, which protects the string from evaluation irrespective of what the string content is.

The first argument of a function, its mnemonic, defines the action to be taken and the way the following arguments have to be interpreted.

Before a function is evaluated, it is necessary that all its argument substrings should have been parsed and evaluated by the scanning algorithm and that there is no argument containing an active function. The parsing within a function is independent of the mnemonic.

This provides the language with versatility, since expressions like the following are perfectly legal:

(i) £(D£(PS, £(RS))S, S1, THIS IS A STRING))

(ii) THIS IS A STRING, IS A STRING STORED UNDER THE NAME S1

If the second statement is typed as input, the following line will be printed:

THIS IS A STRING, IS A STRING STORED UNDER THE NAME S1

which in fact describes the result of the first statement. The mnemonic in this statement is split but this does not affect the final result.

3. Self-referencing facilities: Most of the symbol manipulation systems maintain a group of statements which control the manipulation process and another group containing all strings to which procedures

will be applied. Languages like COMIT, SNOBOL, PANON-1B and AXLE are of this kind.

In such a language, there is no way to interchange control strings and data strings. The result is that a defined procedure cannot be changed at execution time.

In TRAC such a limitation does not exist since both string specifying procedures and data strings are treated in the same way and they can interchange with each other. For example, the expression £(DS, NN, (ABC)) records either a text or a procedure depending on whether the string ABC is a text or a procedure. Subsequently a call to the form NN will produce text or some procedural action. This facility provides the language with self-reference capabilities and allows interaction at execution time.

IIA.14 The "Null" Concept in TRAC Language

The null string represents "nothing" in string terminology but this null concept is more complicated than it sounds. Some cases where the null string appears are: (i) specific string without any character; (ii) an absent or missing string; (iii) absent or missing name; (iv) a specific existing name having no characters; (v) a syntactically correct expression which causes no predefined action; (vi) in a text string, the open space between successive characters — that is, the "space" character; (vii) numerical quantity whose value is zero; (viii) in a card a column with no punches.

In TRAC, the string and its content are two distinct concepts. A string is a form which can hold characters in sequence. If the string contains no characters it is a "null string" but nevertheless it is a perfectly valid string. Thus a string can have a certain

number of characters or no characters at all. If two strings have no characters they satisfy the equality function which is based on a character by character comparison. Since the null string is valid it can be given a name. A name is a string and as such it can be the null string. This behaves exactly like any other name except that it has no visible print indication. For example £(DS,,XXXX) and £(PS, £(C1,)) prints the string XXXX. No space character should be between the two commas in the define function.

The null string is treated uniformly with all other strings. Like any other form, it has a pointer to the beginning and a pointer to the end, but in a null string the beginning coincides with the end and the two pointers point to the same location.

For dealing with absent or missing strings one can write a procedure to search the table of contents for the occurrence of a particular form name, even if this name is the null string. If the string is not in store it is missing. In this case the missing string is considered as a null string. A string might be absent but not null. For example in £(C1, XXXX) there are no parameters and the argument substrings are considered absent and not null.

If a function cannot be interpreted for some reason, for example £(LC, ABC), no action is performed and its value is taken to be the null string. Concatenation of strings is automatic in TRAC and any null string between two non-null strings disappears.

If the result from a numerical computation is zero, it is represented as Ø which is not a null string. Similarly the space character is not a null string.

In general in TRAC the confusion existing in other similar languages about the "null" concept is minimized.

IIA.15 Final Comments

The TRAC language described above is versatile and modular and it can be used in a great variety of applications. The usual manner in which TRAC is operating at various installations is in the implementation of six to ten of its most important primitive functions with the addition of four to six specialized primitive functions or user procedures of local interest. These new functions are designed to be consistent with, and modular to, the TRAC language syntax and philosophy. Examples of such functions are:-

(i) functions to control a data link and automatic dialling equipment;

(ii) functions to command a graphic display unit;

(iii) functions to control mechanical units;

(iv) functions to control paper-tape readers and punches;

(v) functions for digital-to-analog conversion.

Since the ability to define and store procedures always exists, any installed TRAC system can improve itself by including new useful procedures at any time.

IIA.16 References

[1]. C.N. Mooers and L.P. Deutsch, "TRAC, a text handling language", Proc. 20th Natl. Conf. ACM (August 1965).

[2]. C.N. Mooers, "TRAC, a procedure-describing language for the reactive typewriter", Comm. ACM, Vol. 9, No. 3 (March 1966).

[3]. "How some fundamental problems are treated in the design of the TRAC language", Symbol Manipulation Languages and Techniques, Proceedings of the IFIP Working Conference on Symbol Manipulation Languages, ed. D.G. Bobrow, North Holland Publishing Company, Amsterdam.

PART IIB:

TRAC LANGUAGE:

IMPLEMENTATION OF TRAC

PART II: TRAC LANGUAGE

B. IMPLEMENTATION OF TRAC

IIB.1 Introduction

The implemented TRAC interpreter contains the TRAC algorithm and a selected set of primitive functions. The execution of individual functions is accomplished interpretively by consulting a primitive-function code table and transferring control to a subroutine for executing the primitive function. TRAC is a modular language and with the above form of organization, new primitive functions can be added one at a time as they occur to the user without affecting the rest of the system.

The TRAC algorithm was implemented first. It consisted of a main program and four subroutines, namely BEGPAR, SUBCOM, SUBSHA and SUBEND, corresponding to the control characters '(', ',', '£' and ')' respectively. The above subroutines were tested thoroughly using various data to prove that they were operating satisfactorily and when this was accomplished each one of them, after minor modifications, became part of the main program. This was done to minimize execution time and to use core store economically. A comment card with the name of each of the original subroutines still remains, and it can be seen in the listing at the end of this chapter. It shows where the set of instructions for each one of them begins, it facilitates the understanding of the program written, and it recalls the way the TRAC algorithm is implemented.

Each one of the implemented functions was added to the system as a subroutine and a new entry was created in the table containing the

TRAC primitive functions. Each subroutine was tested with a set of TRAC programs. The tests for all implemented functions and the TRAC programs used are referred to in detail in the next chapter, which discusses applications of the TRAC language.

At the final stage the subroutines for the implemented functions were included in the main program for the reasons applicable to the inclusion of the subroutines of the TRAC algorithm. This does not affect the TRAC interpreter since after the table look-up for the function, control is transferred to a numbered statement which is the first of a group of statements which replace the subroutine in the main program. The result after executing this group of instructions is identical to the one the subroutine produced previously.

Other primitive functions can be added easily to the implemented TRAC interpreter. It is advisable to write a subroutine for each added function. This makes it possible to test the new added function without upsetting the rest of the system. A carefully selected set of TRAC programs should be used to cover as many cases as possible. When the new subroutine operates successfully it should become part of the main program as referred to above.

A listing of the input data which were used for testing the TRAC interpreter, followed by their execution, is given at the end of the chapter. This makes it possible when describing various cases that arise during the evaluation of the implemented function to refer to specific input data and the corresponding results. When a message is printed out during execution, either it is of standard length and then it is preceded and followed by five plus characters (+) or it is not of standard length and then only the left five '+' appear.

IIB.2 The TRAC Algorithm

Two stacks (Fig. 1) are required for processing in TRAC: the active string stack where the string to be scanned is copied and the

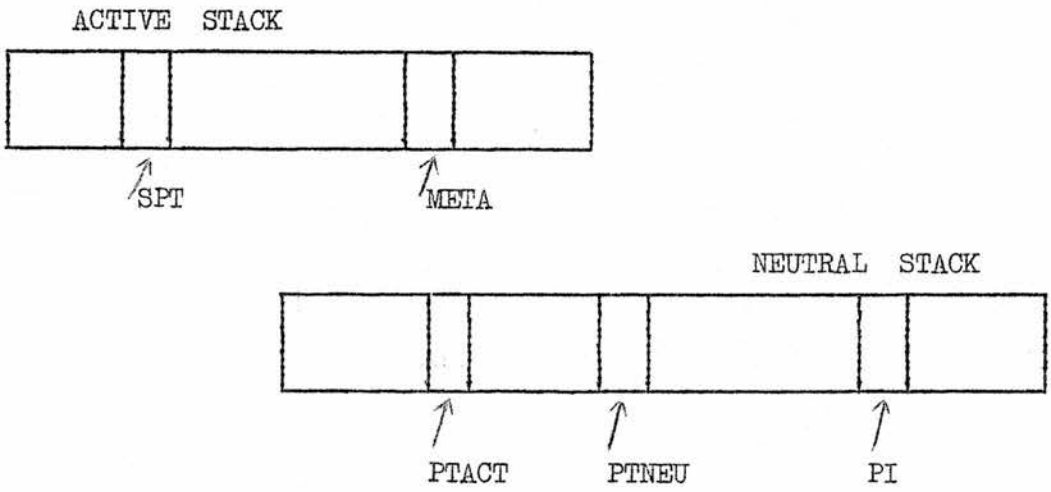


Figure 1

neutral string stack to which the evaluated strings are transferred. The rs function of the idling procedure causes the string to be copied into the active stack followed by evaluative scanning. The neutral string is used as a temporary storage in which evaluated parameters are stored. When all parameters of a string are evaluated its resulting value, if it is not null, passes to the active or neutral stack according to the active or neutral mode of the evaluated string.

A vector of storage is used for both stacks and it is shown in the following figure:

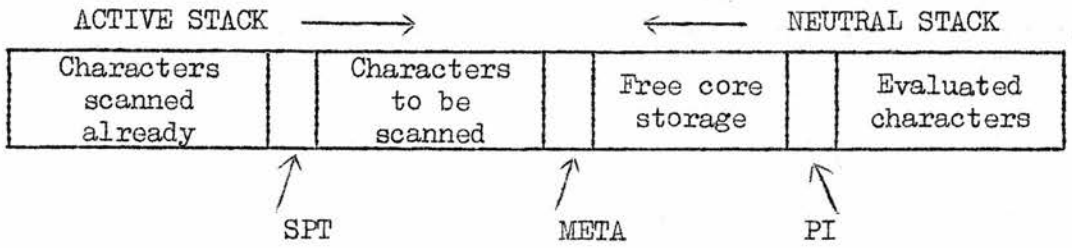


Figure 2

The left-hand side of this vector is used for the active string, named CHAR in the program. The right-hand side is used for the neutral string, named NEU. An equivalence statement allows the same vector of storage to be assigned to both strings.

There are three pointers in the above figure: (i) the scanning pointer, SPT, which points to the evaluated character of the active string; (ii) the end of active string pointer, META, which points to an internal marker MRQUOT — when the marker is reached during the scanning process the TRAC processor is informed that there are no more characters to be scanned in the active string; (iii) the pointer to the next position of the neutral string, PI — this pointer allows an evaluated character to be moved from the active string to the first free place of the neutral string.

IIB.2.1 The idling procedure

The first time the TRAC interpreter is loaded or at the end of a processing cycle the SPT points at the internal marker MRQUOT which shows the end of the active string. The immediate response of the TRAC processor is to load the idling procedure from the table, IDLE, where this procedure is stored, into the active string. The scanning pointer is initialized to the value 1 and the usual scanning process is repeated.

Every time the read string function of the idling procedure is evaluated, the TRAC processor asks for data. If the user does not type any data the second argument of the print string function is null and nothing is printed out. The idling procedure is reloaded automatically and again the same request for typing data is made.

In the present implementation, a facility is built in which allows the user to stop the execution when he has no more data for processing. Each time before the loading of the idling procedure a message appears, +++++ TO END TRAC JOB, TYPE ' IN COLUMN 1 +++++. The user has a choice now. When the request for data appears, he can either enter his data and continue the processing or the quote character. In the latter case, the execution of the TRAC interpreter finishes with an appropriate message.

IIB.2.2 Scanning process

Each character is evaluated by the TRAC scanning algorithm. An ordinary character passes unaltered from the active into the neutral string. A control character causes a certain action to be taken.

The character pointed to by the scanning pointer is examined by the TRAC interpreter. An ordinary character after being evaluated passes into the next free place of the neutral string pointed to by the pointer PI, and 1 is subtracted from this pointer. This makes it possible for the pointer PI to point always to the next free place in the neutral string. The scanning pointer SPT is augmented by one to point at the next character of the active string.

A control character is treated in a different way by the TRAC processor. There is a completely separate procedure to be executed according to which particular control character is pointed at by the scanning pointer of the active string.

IIB.2.3 Left parenthesis

The coding referring to this control character begins after the comment card C SUBROUTINE BEGPAR. A counter is set with an initial value of zero to keep a count of the number of parentheses occurring. One is added to the counter for a left parenthesis and one is subtracted for a right parenthesis. When a left parenthesis is pointed to, if there is no sharp sign before it, the counter is checked. If the content of BEGP is one, all characters following the left parenthesis, even if there are other left or right parentheses or other control characters between them, pass unaltered into the neutral string. This copying from the active into the neutral string continues until the occurrence of the matching parenthesis which makes the value of the counter of parentheses equal to zero.

The character string between this pair of parentheses is in quote mode and it is copied unaltered from the active to the neutral string, only the protecting pair of parentheses being deleted from the string.

IIB.2.4 Comma

The coding referring to this control character begins after the comment card C SUBROUTINE SUBCOM. The comma is a control character, in TRAC, used to separate argument substrings. When passed into the neutral string, it is replaced by an internal marker MARCOM. This marker separates the end of one argument substring from the beginning of the next.

Thus if the character, comma, is needed in a text it should be presented in the quote mode. Then the comma character is protected from evaluation and passes unaltered into the neutral string.

IIB.2.5 Sharp sign

The coding referring to this subroutine begins after the comment card C SUBROUTINE SUBSHA. The sharp sign followed by a left parenthesis is recognized as the beginning of an active function; followed by another sharp sign and a left parenthesis it is recognized as the beginning of a neutral function; and followed by any other character it is considered an ordinary character and passes unaltered into the neutral string. If no evaluation is desirable for a sharp sign, it should be presented in the quote mode.

The beginning of a neutral or active function is marked in the neutral string. In the present implementation, the neutral string is used as a pushdown stack on a last-in, next-out basis for evaluated functions.

The pointers required to access stack components can be stored either within the stack itself or in a separate parameter stack. In

the present implementation, a pointer, PTACT, points to the beginning of the last active function in the neutral string and a pointer, PTNEU, to the beginning of the last neutral function. The rest of the pointers, each pointing to the previous active or neutral function, are stored within the stack. With this organization the neutral string may be represented by Figure 3,

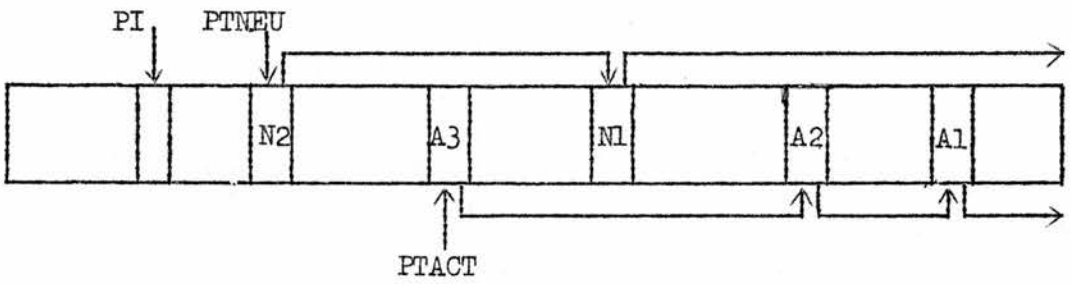


Figure 3

where A1, A2, A3, ... mark the beginning of an active function. The active pointer, PTACT, points to the beginning of the last active function. A1, A2, A3, ... are components of the store and each of them contains a pointer to the previous active function. The pointer of the first function points to a starting value outside the limits of the neutral string. The pointers for neutral functions are stored in the neutral string in a similar way to the pointers of active functions.

IIB.2.6 Right parenthesis

A right parenthesis may be a matching for a left opening parenthesis and then it is deleted. It may be a character of a string in quote mode and then it is copied in the neutral string unaltered. It may be the closing parenthesis for a function.

The right parenthesis which denotes the end of a function is deleted. The TRAC processor then retrieves the last evaluated function from the top of the neutral string stack.

A comparison of the two pointers, PTACT and PTNEU, is made to identify the kind of the function. The pointer with the smaller content is the pointer of the last function. In the figure above PTNEU is less than PTACT. Thus the TRAC interpreter is informed that the last added function is a neutral one. The content of this last function is at the top of the stack between the two places pointed at by the pointers PTNEU and PI.

The first two characters after the pointer of the function are now retrieved and a look-up of the table containing the implemented functions takes place. This table is initialized in the Block data subprogram. Each time a function is implemented a new entry is created in the table.

When the mnemonic of the function matches one of the entries of the table control is transferred to the appropriate coding for the function. When there is no match, an error message for the function is printed followed by the function itself for inspection. The function is considered null-valued, it causes no action and it is deleted from the neutral string. The function with no predetermined action is considered as null and does not upset the system which allows the means for self-recovery from trivial errors.

Another entry in the table with the implemented forms shows whether the function is null-valued or not. This is valuable for the TRAC processor since much of its subsequent action depends on this information.

IIB.3 Evaluation of a Function

When the mnemonic of the scanned function matches one of the entries in the table of function names, then action for the matched

function takes place and control is transferred to the appropriate set of instructions. A computed GO TO statement is used for this transfer. This GO TO contains as many exits as the number of implemented functions plus one. The last exit allows the system to recover from errors occurring in typing the mnemonic of a function or even in cases where, although the mnemonic of the function is correct, the function is not present in the implementation.

All parameters for the evaluated function are retrieved from the top of the neutral stack. The pointer, PTACT or PTNEU, shows where this function, active or neutral, begins and the pointer PI shows the end of the function. When the function is a null-valued one and its corresponding set of instructions has been executed and thus the evaluation has finished, the function is deleted from the top of the neutral stack. The pointers for the deleted function are reset. The pointer PI is set to the beginning of the deleted function and the PTACT or PTNEU pointer is set to the beginning of the function before the deleted one.

When the function has a value, there are more factors to consider. The function value has to be inserted at the top of the active or neutral stack according to the mode, active or neutral, of the evaluated function. The order of evaluation and deletion for a particular function with value depends on the particular function and this will become clear when describing each implemented function with value.

At the moment, only the way this value is inserted on the top of the corresponding stack is described. It is rather simpler for the value of a neutral function. This value is moved, character by character, to the top of the neutral stack and occupies the free core storage of the TRAC processor. The pointer PI is always set to the next free place of the neutral stack. Thus any number of characters can be inserted on the top of the neutral stack, if there is free core

store in the processor. If there is no free core left, a reference is made to a subroutine CLEARP which takes care of this matter. This subroutine will be described fully in a later section.

If the evaluated function is active, its value is inserted on the top of the active stack. Thus the next instruction to be executed when execution of a given instruction has been completed is always found at the top of the active string stack. The scanning pointer, SPT, is always pointing at the top of the active stack. The core storage to the left of this pointer is considered free since it consists of evaluated characters.

If the number of characters of the inserted value is less than the value of the SPT, the active string stack is moved to the left and some more core is added to the free core storage which is on the right of the pointer META. This implementation adds to the efficiency of the system since it allows the strings supplied by the user to be copied in linear fashion, readjusted, recopied or deleted. In this way most of the garbage collection is done during processing.

If the number of characters of the inserted value is equal to the value of the SPT, the active stack is not moved.

If the number of characters of the inserted value is greater than the value of the SPT, the unevaluated active stack is moved to the right before the value of the active function is inserted on the top of the active stack. This movement of the active stack takes place to provide enough core store for inserting the value of the active function on top of the active stack. It is possible to move the active stack only when there is enough free core storage left; otherwise a reference is made to the subroutine CLEARP.

In all three cases the scanning pointer is reset to the value, one, to point at the first character of the inserted value on top of the stack. This is the first character to be evaluated during the next processing cycle.

Thus the implemented system is completely general and allows the

user to insert any number of characters on top of both stacks, provided, of course, that there is enough core storage. This is the only limitation.

IIB.4 General Remarks

The rs function of the idling procedure causes any input string to be copied into the active stack and then to be evaluated by the TRAC scanning algorithm, which uses the neutral stack as a temporary storage.

TRAC is a functional language which executes instructions, which are represented by the system's functions, by replacing the string of characters constituting the instruction by its value, if there is one, and sometimes causing side effects in the TRAC memory, in the input/output medium or in the mode of operation of the TRAC processor.

A very important feature of the TRAC language is that there is a clear distinction between strings which are to be treated as instructions and strings which are to be treated as data. This is accomplished by placing strings to be executed as instructions in the active string and strings to be used as data in the neutral string. The arguments of an instruction are initially in the active string and they are treated as instructions when they are scanned. Thus nested instructions can be executed by the TRAC processor.

The value of any neutral function is treated as data and it is placed on the neutral string. Thus it is possible, using neutral functions, to avoid copying the character string first in the active string and moving it character by character to the neutral string afterwards. In the following sections the implementation of each included function is described and the subroutine CLEARP, which is used for the garbage collection of the implemented system, is examined in detail.

IIB.5 The Read String Function

This function has a value and this value is inserted in the active or neutral string according to the mode, active or neutral, of the read string function. When such a function is detected by the TRAC interpreter the function is first deleted from the neutral string before the value is inserted and the pointers PI and PTNEU or PTACT are reset to the next free place of the neutral string and to the beginning of the previous neutral or active function.

Thus if the function is neutral, its value can immediately replace the rs function on the neutral string. If the function is active, its value will be evaluated in the active string and the evaluated string value will replace the old rs in the neutral string.

The input character string is read in card format. An array of 80 characters, VAL(80), is kept as the card image in memory, and from there the input string is transferred to the active or neutral stack. A counter, CARD, is kept for counting the lines that the input string contains. When the first character of the first line is the quote character "'" the execution of the particular TRAC job finishes, since this is the option that the system provides the user to prevent the loading of the idling procedure and so stop the processing. If the above test fails all read characters are examined for an occurrence of the terminating character, which is the quote character. If none of the 80 characters read matches the terminating character the whole line is put in the top of the active or neutral string according to the mode of the function. Auxiliary pointers make it possible to insert new lines of input one at a time. When a new line of input is inserted in the active string, its first character follows the last one of the previously inserted line and its last one is followed by the unevaluated characters of the active string, which are moved to the left or to the

right to provide the necessary storage for the inserted line of input.

A much simpler process is needed when the rs function is neutral, since then each line of its value is inserted on the top of the neutral stack following the one that has been inserted previously. If the terminating character is not found in one line of input the TRAC interpreter asks for another line of input. It reads and inserts it on the top of the appropriate stack. This procedure of reading and inserting lines of input continues until the occurrence of the terminating character.

The terminating character is deleted from the input string but all other preceding characters are treated as previously. A switch, S1, is set on and then control is not transferred back to the read statement. The scanning process continues from the character pointed by the SPT and this is the exit from the function rs.

A missing terminating character when there is no more input to be read in causes the print-out of a message "END OF DATA WITHOUT TERMINATING CHARACTER" when the job is submitted as a background job. In the interactive form the processor asks for more data.

The way of implementation of the rs function allows any number of lines to be read in and form the input character string which can have any length, provided that there is enough core storage available in the TRAC processor.

IIB.6 The Print String Function

In the present implementation the character string contained between the comma after the mnemonic of the function and the right parenthesis, which corresponds to its end, is considered as the second argument and it is printed out. (See line 52 of the input and the

corresponding line 186 of the results.)

When the second argument of the ps function is the null string, nothing is printed out and the message 'THIS PS FUNCTION HAS NULL STRING' appears on the output medium. (See line 117 of the input and the corresponding lines 518-525 of the results.)

Another message 'ERROR FUNCTION' is given when the mnemonic of the function is not followed by a comma. The function is printed out for inspection and the scanning continues from the next instruction of the active string. (See line 162 of the input and the corresponding lines 747-751 of the results.)

In all three cases, the print string function is deleted from the neutral string and then the scanning continues from the next function of the active string.

IIB.7 The Define String Function

A vector of core storage named FSTORE is used for storing the defined forms. The following figure is a representation of FSTORE:

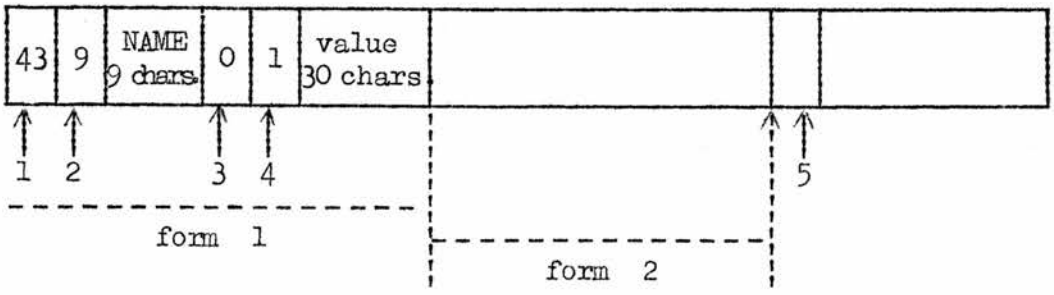


Figure 4

1. Pointer to the last character of the value.
2. Indicator of the number of characters the name contains.
3. Segment gap indicator.
4. Pointer to the first character of the value.
5. End of store marker.

Before executing the define string function the TRAC interpreter examines the character which follows the mnemonic of the function. If this character is not the internal marker, MARCOM, which replaces the character, comma, a message is printed out. The form is not defined but the function is typed out for inspection and control is transferred to the next instruction on the active string. (See line 121 of the input and the corresponding lines 545-548 of the results.)

The scanning on top of the neutral string continues for the occurrence of a second comma. The string contained between the two commas is the name of the form and a counter, NONAME, contains the number of characters of the form name.

All characters following the second comma until the end of the function are the value of the form. This form, name and value together with some auxiliary pointers, is to be stored in the free space of the vector FSTORE. The first free place of FSTORE is marked with the internal marker MEMPTY. Initially the first place of FSTORE is marked with this marker. This informs the TRAC processor that FSTORE is empty and no form is stored there for the moment. When one or more forms are stored in FSTORE, the marker, MEMPTY, is moved to the right to the first free place of FSTORE.

A search is made in FSTORE before storing any form. A pointer, PTSTOR, is used for the scanning of the store of forms. This pointer is set initially to the first place. When the store contains no forms the marker MEMPTY is found there.

The way the first form is stored is now described. This will help in understanding how the search for the free space is carried out when there are other forms already stored in FSTORE.

Consider as the first form to be stored the one in line 90 of the input. Since the marker MEMPTY is found in the place pointed at by PTSTOR, the processor assumes there is free space for storing the form.

PTSTOR points to the next place and there the number of characters of the name of the form is stored. This number is an indicator of how many places the actual name of the form occupies. The name of the form is retrieved from the neutral string and it is stored character by character in the next free places pointed at by the pointer PTSTOR.

In the next place an integer, used as a segment gap indicator, is stored. If there are no segment gaps, zero is stored in this place. Another integer, pointer to the value of the form, follows. This pointer is set to the first character of the value initially.

The value of the form is now moved character by character from the neutral stack and it is stored in the next free places pointed to by PTSTOR. The storing of the value finishes when the pointer, LOC4, used for retrieving characters from the neutral string becomes equal to the pointer PI pointing to the free top of the stack.

The pointer PTSTOR is set to the next place and there the internal marker MEMPTY is stored to show the beginning of the free FSTORE. In the previous location of MEMPTY another pointer is set to the last value of the form. The content of the first place of the form is added to a counter, TOTAL, which keeps the total number of characters stored, excluding the marker, MEMPTY.

The evaluation of the function finishes at this point and after the initialization of pointers and counters, control is transferred to the main process. Then the define string function is deleted from the neutral stack and execution continues with the next instruction from the active stack.

When a new form is defined and several are already stored, then the store of forms is scanned until the marker MEMPTY is found. The processor can detect during this operation whether another form, with the same name, already exists in store. This is accomplished by comparing the names of both forms. The one form is that pointed at by

PTSTOR and the other is on the top of the neutral stack.

The above test is executed only when the two names contain the same number of characters. Two identical names cause the interpreter to delete the old form from storage by shifting to the left all forms defined after the deleted one, and the marker MEMPTY. Thus the gap created after the deletion of a form does not remain but is filled with some of the shifted characters. The new form is inserted now in the free core of FSTORE. (See lines 165-168 of the input and the corresponding lines 757-787 of the results.)

When there is no form stored with the same name, the scanning process continues until the marker, MEMPTY, is found indicating the beginning of the free store, if, of course, FSTORE is not full yet. It should be stressed here that the way FSTORE is organized contributes a great deal to the efficiency of the system. Pointers which are kept in FSTORE make it possible for the processor to scan the store form by form rather than character by character.

The processor inserts the whole form name, value and pointers character by character in the free core storage of FSTORE. Every time a character is moved in FSTORE, the pointer, PTSTOR, which is always set to the next free place, is compared with DIM, which contains the dimension of FSTORE. If the value in PTSTOR is greater than the one in DIM a message, 'THERE IS NO PLACE IN STORE AVAILABLE FOR FORMS', followed by the function causing the overflow of FSTORE, is printed out. A switch is set on and the TRAC interpreter responds with the same message to any further attempt to define forms. The implemented system provides facilities for recovering from an FSTORE overflow. The da function can be used to clear FSTORE and to set off the above switch. Thus it is again possible to store forms in the store that is now free. (See lines 127-137 of the input and the corresponding lines 586-626 of the results.)

The concept of null is introduced in this implementation of the ds function.

Forms with null names can be defined, stored and used in subsequent processing in a way similar to the one used for forms having names with any number of characters. The only difference is in storing such a form. In the second place of a null-named form, is placed the number zero to indicate that there is no name string following. (See lines 118-120 of the input and the corresponding lines 525-542 of the results.)

Forms with null values are treated in the same way as forms having value. A form with null value has the pointer to the last character of the form pointing now to the pointer to the first character of the form value. Since the form has no value the last pointer is now pointing to the next place of the null-valued form. (See lines 6-9 and 17-20 of the input and the corresponding lines 8-27 and 57-82 in the results.) It is even possible to combine null name and null value and to define such a form. (See line 169 of the input and the corresponding lines 788-799 of the results.)

IIB.8 The Segment String Function

The ss function is implemented in a way similar to the one for the ds function. The reason is that the ss function can be thought of as a more general procedure for redefining an existing form. The additional feature is the addition of segment gaps in the value of the form. Thus some of the operations used in the ds function are used here with minor modifications when necessary.

Procedures described in the previous section are simply mentioned briefly here and references to the tested cases are given. A full description is given for new operations introduced in this function.

The test for comma after the mnemonic is repeated for the ss function. (See line 146 of the input and corresponding lines 662-669 of the results.)

When the test for the first comma is successful, the scanning of the ss function continues until a second one is found and then the string between the two commas is identified as the name of the form to be segmented. The pointer, LOC4, used to retrieve characters from the neutral stack, might reach the value of the top pointer PI before the second comma is found or immediately after that. Then the processor prints out a message "THERE ARE NO ARGUMENTS IN THAT FUNCTION" followed by the function itself, counters and pointers are set to their initial values and control is transferred to the next instruction. (See lines 147-148 of the input and the corresponding lines 670-676 of the results.)

If there are argument substrings following the name in the ss function, FSTORE is scanned to find the form with the same name. The message 'THERE IS NO SUCH FORM IN STORE TO BE SEGMENTED', followed by the ss function are printed out when the form is not found. Counters and pointers are initialized and control passes to the next instruction. (See line 170 of the input and the corresponding lines 799-804 of the results.)

A match of the names causes a segmentation of the form if there are substrings in its value matching the arguments of the ss function. Such substrings are replaced by internal markers of the same ordinal value as the matching argument of the ss function. A marker of ordinal value n (>0) is given the internal value MEMPTY-n. The marker MEMPTY holds the value 32757. Thus the internal value of any marker can be found given its order, for example the marker of ordinal value 4 is represented internally with the value 32753.

The special characters = ' , / -) * £ & + (. b (blank) are

represented respectively by the numbers 32320, 32064, 27456, 24896, 24640, 23872, 23616, 23360, 20544, 20032, 19776, 19264, 16488. Thus for general applications this implementation is efficient in that it provides markers up to ordinal value 436. For special cases where more arguments appear and thus more markers are needed, an addition to the program should be made to exclude the values of the above special characters. Then more than 30000 markers would be available. If the maximum number of markers, of INTEGER*2 size, is needed the values of letters and numbers should be excluded as well. The largest value is the one of number 9, -1728.

The form is scanned once for each argument substring of the ss function. When a substring of the value matches an argument of the function, its first character is replaced by the internal marker of the appropriate ordinal value and each of the remaining characters is replaced by the marker FORGET. The segment gap indicator is updated to contain the maximum ordinal value of the inserted markers, a non-segmented form contains zero in this place. There is no limit to the number of characters each argument may contain.

When the scanning of the form for the last argument of the ss function finishes, the segment gap indicator is tested. If it still contains the number zero, the form is not segmented. The message, 'NON SEGMENTATION OF THAT FORM HAS OCCURRED', followed by the ss function, is printed out and after the usual initialization process control passes to the next instruction. (See lines 171-172 of the input and the corresponding lines 809-820 of the results.)

A segmented form is transferred, like a new, defined form, character by character to the free store of FSTORE with all FORGET characters deleted. Finally all forms after the segmented one are shifted to the left, the old form with the FORGET markers is deleted and the segmented one remains with the internal markers as segment gap

indicators.

Every time a character of the segmented form is transferred to the next free place of FSTORE, the used pointer, TOTAL, is tested. If it exceeds DIM, where the dimension of FSTORE is kept, there is no free store left in FSTORE. A message 'FORM CANNOT BE SEGMENTED PROCESSOR FULL', and the function ss are printed out and after initialization control is transferred to the next instruction. (See lines 154-158 of the input and the corresponding lines 712-726 of the results.)

The implemented system can recover when such a case is met. The da function can be used to empty FSTORE again and make it available for storing forms. (See lines 159-161 of the input and the corresponding lines 728-743 of the results.)

The null concept is fully implemented. A null-named form can be segmented. (See lines 44-48 of the input and the corresponding lines 189-227 of the results.) Null arguments in a ss function do not create segment gaps, but they do not upset the system. (See lines 149-153 of the input and the corresponding lines 685-711 of the results.)

IIB.9 The Call Function

This function causes the internal markers of a form to be replaced by the argument substrings of the call function. There is no limit to the number of characters each argument may contain except the core storage limitations. Each marker is replaced by the argument corresponding to its ordinal value. A marker can occur any number of times in the value of a form.

The following operations are executed during the evaluation of a call function:-

(1) The function is scanned and various tasks are carried out to

test its coding. When the coding is correct, name and arguments can be retrieved for further processing.

(2) The store is scanned until the form with the name given in the call function is found. If such a form exists, its value is scanned for occurrences of internal markers which are replaced by the function arguments.

(3) The call function is deleted from the top of the neutral stack, and is replaced by its value if the function is in neutral mode. If the mode is active the value is placed on top of the active string.

Evaluation begins with the test for a comma after the function's mnemonic. A failure causes the print-out of a message followed by the function which is printed for inspection. Subsequent steps are similar to the ss function. (See line 61 of the input and corresponding lines 190-191 of the results.) When the above test is successful, the scanning continues for the occurrence of the second comma. This makes it possible to identify the function name as the string between the two commas. There is a difference from the ss function here. The call function may lack argument substrings and still be a valid function. A switch, FF1, is set on when lack of arguments is detected and a message is printed out. (See lines 62 and 64 of the input and the corresponding lines 193 and 200 of the results.)

The FSTORE is scanned for the form with the name given in the call function. If no such form exists, the appropriate message and the function are printed out and the usual procedure for initialization is repeated. (See line 66 of the input and the corresponding lines 210-211 of the results.)

If the name is found and the form exists in FSTORE, then the value of the identified form is scanned, character by character. Any existing marker is identified by its ordinal value and has to be replaced by the appropriate argument. The segment gap indicator of the

form gives the maximum ordinal value of existing markers. Thus each character is compared with each one of the allowable values that markers may have in the form examined.

An ordinary character of the value is just moved to the next free place of FSTORE. Wherever a marker appears and the switch FF1 is on, the message 'THE CL FUNCTION HAS NO ARGUMENTS MARKERS REPLACED BY NULL' is printed out and in fact the null string replaces the marker in the value string of the call function. This value is formed at the free core of FSTORE. (See lines 62 and 64 of the input and the corresponding lines 194-197 and 201-204 of the results.)

If the above switch is off, the corresponding argument has to be retrieved from the neutral stack to replace the marker. If the number of arguments is less than the maximum ordinal value of markers then markers with greater ordinal value than the number of arguments are replaced by null in the value string and a message is printed out wherever such a marker occurs. (See line 67 of the input and the corresponding lines 213-216 of the results.) If the number of arguments is greater than the maximum ordinal value of markers, the arguments with order greater than this value are just ignored. (See line 68 of the input and the corresponding line 218 of the results.) If there are no markers in the value of the form, the segment gap indicator is zero, the value of the call function is a copy of the form value and there is no movement of such a value to the free core of FSTORE. A warning message is printed out. (See line 65 of the input and the corresponding lines 206-208 of the results.)

Wherever a character is placed at the next free place of FSTORE, the pointer PROS must be not greater than DIM where the value of the dimension of FSTORE is kept. If the value of the pointer is greater than the dimension of FSTORE, the form storage is full and processing cannot continue. A message and the corresponding call function are

printed out. (See lines 28-33 of the input and the corresponding lines 111-129 of the results.) If it is desirable to use FSTORE for further processing the da function is used. (See lines 34-36 of the input and the corresponding lines 130-147 of the results.)

Various cases of null have been implemented and thus it is possible to call (i) a null-named form (see lines 47-48 of the input and the corresponding lines 193-205 of the results); (ii) a null-valued form (see lines 19-20 of the input and corresponding lines 71-84 of the results); (iii) a null-named, null-valued form (see line 169 of the input and the corresponding lines 793-798 of the results). A call function might have some null arguments. (See line 15 of the input and the corresponding lines 42-50 of the results.)

When the value of the call function is evaluated in FSTORE, the function is deleted from the top of the active stack. This deletion is followed by the insertion of the function value on top of the neutral or active stack according to the mode of the deleted function.

The procedure for the insertion of the value is similar to the one used for the rs function. There is a difference, however, in the way this value is transferred. Since in the call function the value is already evaluated before the insertion process begins, this value is copied on top of the appropriate stack, character by character, until the whole string of the value is exhausted. There is no need to use the card format during this insertion.

IIB.10 The Remaining Implemented Functions

The other implemented functions are described briefly. The way each of them was introduced as a subroutine at the beginning and subsequently as a set of instructions added to the main program is

identical with the one for the main TRAC primitive functions.

The basic principles of the implementation are clear by now. Thus instead of a detailed description for each of the implemented functions, whenever possible references to the test of the TRAC interpretation are given. Thus the input data and the results they produce are used to clarify various cases that can be met when each of these functions is evaluated.

1. The trace-on/trace-off functions

The flow of execution is traced by the print-out of each function, before its evaluation from the top of the neutral stack. A switch, set on when the trace-on function is evaluated, causes this print-out to start. The same switch, set off when the trace-off function is evaluated, causes this print-out to stop. (See lines 69-76 of the input and the corresponding lines 229-331 of the results.)

2. The delete-all function

The da function is used to delete all existing functions from FSTORE. When this function is evaluated, the internal marker MEMPTY is moved to the first place of FSTORE and this is interpreted as free FSTORE from the TRAC processor.

It can be used either when stored forms are not needed any longer or for recovering from an overflow of FSTORE. (See lines 77-78 of the input and the corresponding lines 335-392 of the results for the former case and the lines 28-35 and 111-140 respectively for the latter.)

3. The list names function

When this function is executed, a listing of the names of stored forms is printed out with each name preceded by the second argument of the LN function. When either the name of the last form has been printed out or FSTORE is empty when the function is executed, the message 'THERE ARE NO FORMS IN STORE' is printed out. (See line 75 of the input and the corresponding lines 290-315 of the results, and also

line 78 and the corresponding line 351 of the result when FSTORE is empty)

It can be seen, in line 75, that the function £(LN COMMA MISSING) is coded the wrong way and thus a message followed by the function appears in lines 306-308 of the results. In the same line, 76, the second argument of the form £(LN,) is null and only the form names are printed out, lines 309-315.

In more detail, the cases where null strings are met are the following:

(a) Form with null name, but the second argument of the LN function not null. This argument is printed out followed by null. (See line 292 of the results.)

(b) Second argument of LN function null but form name not null. The name is printed preceded by the null string. (See lines 309-315 of the results.)

(c) Both second argument and form name null. The null string is printed out. In line 310 X2 is printed out as the first name. Although X2 is the second form in FSTORE and the first is the null-named form nothing — null string — is printed out for the null-named form since the second argument of the LN function is null as well.

4. The print form function

This function causes the print-out of any specified form if such a form exists in store. If the coding of the pf is not correct, a message and the function are printed out. (See line 173 of the input and the corresponding lines 823-825 of the results.) If either the store is empty or there is no form with the specified name the appropriate message followed by the function is printed out. (See lines 77-78 of the input and the corresponding lines 342-352 of the results for the former and lines 174 and 832-834 respectively for the latter.) If the form is not segmented its value is printed out as it

is stored in FSTORE. (See lines 8-9 of the input and the corresponding lines 22-28 of the results.) If the form is segmented, an indicator (here the ampersand is chosen) is printed out for the segment gap followed by a number equal to its ordinal value. (See line 74 of the input and the corresponding lines 262-270 of the results.) The last two lines 269, 270 of these results refer to the function £(PF,) where the name of the form is the null string. When the value of the form is null the appropriate message, followed by the function, is printed out. (See lines 6-7 of the input and the corresponding lines 8-20 of the results.)

The process for the print-out of any specified form is the following. A vector, LINE(80), of a card format is kept as a buffer area in store. The first stored character is a blank, used as a control character. Whenever this buffer area is filled with characters it is printed out as a line. Any character of the form value, unless it is an internal marker, is moved to the next free place of this buffer. The segment gap indicator of the form makes it possible to test each character with all allowable values for internal markers. Whenever a marker occurs in the value of the form, an ampersand is moved to the buffer area. Since input/output operations occur with transmission of characters rather than numbers, the ordinal value of the marker, which must follow the ampersand in the print-out, has to be transformed. The integer, which represents the ordinal value of the marker, is transformed to an array of characters where each character of this array is a digit of the numerical value.

5. The equal function

The equal function is a conditional one and is used for alternative transfers of control or construction of different strings. The second and third argument substrings are tested and either they are equal and the value of the equal function is the fourth argument or

they are not equal and the value of the equal function is the fifth argument.

If the coding of the function is not correct, a message and the function are printed out. The number of expected arguments is always five, although some of them may be null strings. (See lines 180-184 of the input and the corresponding lines 872-908 of the results.) The value of the function may be the fourth argument. (See lines 188-191 of the input and the corresponding lines 927-957 of the results.) In line 189 of the input the second and third arguments are null strings and are considered as equal. In line 190 the second and third arguments are null-valued functions and they are replaced by null substrings after their evaluation. In line 191 the fourth argument is a call to a form with null value and thus the value of the equal function is the null string. Similar cases are included (see lines 192-195 of the input and the corresponding lines 958-996 of the results) when the value of the function is the fifth argument.

As already mentioned, the value of the equal function is its fourth or fifth argument. Two different cases should be considered:

(a) The value argument after its evaluation results in a character string. This character string is copied from its specific place between two markers standing for commas to the top of the neutral or active stack, according to the mode of the equal function.

(b) The value argument may contain one or more functions which were presented in the quote mode and thus protected from evaluation. (See line 36 of the input and the corresponding lines 140-147 of the results.) The fourth argument is presented in the quote mode (£(CL, TESTEQ)), thus the value of the function is another function which replaces the equal function on top of the appropriate stack. In the first case the value of the equal function may be a null string. In the second case, there is always a value for the equal function.

This is one or more functions and these can be either functions with value or null-valued functions.

IIB.11 The Garbage Collection

It has been mentioned in section IIB.2 that one vector of storage is used for both active and neutral stacks. This vector of storage is usually referred to as the TRAC processor. It is evident that for the continuation of processing some free core storage is needed between the two stacks. When the two pointers, META, to the end of the active string, and PI, to the end of the neutral string, become equal the two strings, active and neutral, meet each other and there is no free core store left for further processing. This is detected by the TRAC interpreter and processing does not continue unless certain action is taken to provide some free store. This action is usually referred to as the garbage collection operation.

In the implementation of the present system most of the garbage collection is done during processing by keeping various pointers and moving strings in a linear fashion. This applies to both active and neutral stack. Whenever the value of an active function is moved to the top of the active stack the remaining unevaluated characters of the active string are moved to the left or to the right. When the number of characters of this value is less than the number of characters that have already been evaluated, the active string is moved to the left and some more free store is added to the existing one. Every time a function is evaluated, it is deleted from the top of the neutral stack. Thus some more core store is added again unless the deleted function has a value and the number of characters of this value is greater than or equal to the number of characters of the deleted function.

There are, however, some cases when more systematic garbage collection operations have to be carried out. There are two operations that might cause an overflow of the TRAC processor:-

(1) The transfer of any evaluated character from the active to the neutral string.

(2) The value of a function, active or neutral, inserted at the top of the corresponding stack.

In the first case, the system provides the means for recovering. Any time a character is evaluated, it is transferred from the active to the neutral stack and subsequently one is subtracted from the pointer, PI, which is compared with the pointer, META. It can be seen that if after one transfer as above the two pointers are equal, there is always some free store at the left of the unevaluated active string. This free store is created by the evaluated characters. When META equals PI, the system transfers control to the subroutine CLEARP. Then the active string is moved to the left in the place of the evaluated characters, some free core storage is created and processing continues normally. Consider the example lines 257-270 of the input and the corresponding lines 1101-1145 of the results. This example demonstrates the way the system recovers and processing continues. Each time the two strings meet each other, the active string is moved to the left and thus some free core storage is created which allows the movement of the evaluated characters from the active to the neutral stack. The input contains a string of 960 characters. (In this implementation the capacity of the processor is 1000 characters.) The garbage collection operation is carried out 10 times and by 10 successive movements of the active string to the left the input string is evaluated and results are printed out.

In the second case, recovery is not always possible. The processor is full when the number of characters of the value of the

function, active or neutral, is greater than or equal to the number of the scanned characters plus the number of free store places. When the function is neutral, the test is similar to the first case; every time one is subtracted from PI its value is compared with META and when they are equal, control is transferred to CLEARP. When the function is active, it is only when the active string is moved to the right that the pointer META is compared with PI and when META is greater than or equal to PI, control is transferred to CLEARP again. This subroutine causes a message PROCESSOR FULL EXECUTION STOPS and the contents of the processor to be printed out, only when SPT = 1 otherwise a movement of the active string to the left takes place and processing continues.

Some cases are demonstrated in the test for the interpreter. (See lines 198-214 of the input and the corresponding lines 1003-1023 of the results for an overflow caused by an active RS function; lines 216-230, 1025-1045 respectively for one overflow caused by a neutral function; and similarly lines 232-247, 1046-1069 for an overflow caused by an active cl function; lines 249-255, 1071-1099 for an overflow caused by a neutral cl function.)

IIB.12 Final Comments and Suggestions for Further Extension

The present system can be extended by implementing more functions from those existing in the TRAC language or even by writing new specialized functions. Each function should be added to the system in the form of a subroutine and after being tested successfully with various data it can be included in the main program.

Some of the necessary changes to the program for each added function are the following:-

1. One new entry is added to the function table FUNC. This entry must contain the mnemonic of the added function and an indication of whether this is a function with value or not. In the former case the indication is the literal ' ' (blank) and in the latter the character 'N'. The way in which this table is organized can be seen in the BLOCK DATA subprogram.

2. One is added to the counter of the implemented functions, N1. (N1 is equal to 11 in the present system.)

3. A new exit is added, before the last one, in the computed GØ TØ statement. This computed GØ TØ is used for transfer of control to the appropriate function. The last exit of the statement corresponds to any mnemonic which is not included in the function table. When a function is retrieved and its mnemonic is not included in the table, a message "ERROR CODED FUNCTION DELETED" and the function are printed out and after the deletion of the function processing continues normally. (See lines 83-84 of the input and the corresponding lines 366-373 of the results.)

When the added function has null value, whenever its mnemonic is retrieved during the scanning process the corresponding exit of the computed GØ TØ statement transfers control to the appropriate set of instructions. When these instructions are executed and the function is evaluated, control returns to the main processing and the next instruction to be executed is the same as for the other already implemented functions, the ps function for example. This instruction is, in fact, the first of a set of functions used to delete the evaluated function, active or neutral, from the top of the neutral stack.

When the added function has a value, some more factors have to be considered. The corresponding set of instructions includes the following operations:-

(a) the evaluation of the function value;

(b) the insertion of the value on top of the active or neutral stack, according to the mode of the function;

(c) the deletion of the evaluated function from the top of the neutral stack;

(d) the return of control to the main processing as in any other of the implemented functions, for example cl function.

The order and the way these operations are carried out might vary for different functions. Any time a value is transferred to the top of the active stack, the remaining unevaluated characters are shifted left or right and the corresponding pointers are readjusted. If the value is transferred to the top of the neutral stack no shifting takes place and only the pointer to the top of the stack is updated.

Whenever a value of a function is transferred to the active or neutral stack or any character is evaluated and moved from the active to the neutral string, the pointers META and PI should be compared. If META is greater than or equal to PI, control passes to the subroutine CLEARP for the garbage collection.

The implemented system allows the print-out of various messages at execution time. Thus the user is notified by the TRAC interpreter whenever an error is detected. Other messages appear simply to enable the user to follow and understand as far as possible some of the actions taken by the interpreter during the processing. If desirable — to save program storage space or to produce a simpler print-out — some or all of these messages may be dispensed with by removing from the program all WRITE statements referring to them. Such statements can be detected easily in the listing of the program since they are followed by FORMAT statements containing the actual message between the plus characters /FORMAT(+++++message+++++)/. If this were done, only the

results and any remaining messages would be printed out.

Finally it should be noted that although the action of the TRAC interpreter in the evaluation of the nested functions is highly recursive, the computer program is not. That is, none of the subroutines for the implemented functions is entered twice. The recursive action of the interpreter is achieved by using the TRAC scanning algorithm and transferring the results of the scanning process in the neutral stack before any nested function is evaluated. Evaluation begins always from the top of the stack and the first evaluated function in a nested expression is the innermost one.

IIB.13 Test of TRAC Interpreter

. . . pages 113-120

```
**************************************************************************** 1
                          TEST OF TRAC INTERPRETER                            2
**************************************************************************** 3
                          INPUT DATA                                          4
£(TN)'                                                                        5
£(DS,NULL VALUE FORM1,)£(DS,NULL VALUE FORM2)£(PF,NULL VALUE FORM1)£(PF,NULL VAL  6
UE FORM2)'                                                                    7
£(DS,NULL VALUE FORM1,FORM1 IS REDEFINED IT IS NOT NULL ANY MORE) £(PF,NULL VALU  8
E FORM1)'                                                                     9
£(DA)'                                                                       10
£(DS,FORM1,    FIRST DEFINED FORM                                    )       11
£(DS,FORM2,2ND DEFINED A FORM B     C                                )       12
£(DS,FORM3,3RD DEFINED FORM                                          )       13
£(SS,FORM2,A,B,C)'                                                           14
£(CL,FORM2,AAA,£(DS,B,BBB)£(PS,B),BBB,,CCC)'                                 15
£(DA)'                                                                       16
£(DS,STRING1 WITH NULL VALUE,)'                                              17
£(DS,STRING2 WITH NULL VALUE)'                                               18
£(CL,STRING1 WITH NULL VALUE)'                                               19
£(CL,STRING2 WITH NULL VALUE)'                                               20
£(EQ,£(CL,STRING1 WITH NULL VALUE),£(CL,STRING2 WITH NULL VALUE),STRINGS EQUAL S  21
INCE BOTH HAVE NULL VALUE,ERROR)'                                            22
££(DA)'                                                                      23
£(DS,FORM1,    FIRST DEFINED FORM                                    )       24
£(DS,FORM2,2ND DEFINED A FORM B     C                                )       25
£(DS,FORM3,3RD DEFINED FORM                                          )       26
£(SS,FORM2,A,B,C)'                                                           27
£(CL,FORM2,   CARD1                                                          28
          CARD2                                                         ,      29
             CARD3                                                           30
                  CARD4                                                 ,      31
                  CARD5                                                       32
        CARD6                                                         )'      33
£(DA)'                                                                       34
£(DS,TESTEQ,2ND AND 3RD ARGUMENTS OF EQ FUNCTION ARE EQUAL SINCE BOTH NULL)'      35
£(EQ,,,(£(CL,TESTEQ)),ERROR FUNCTION)'                                       36
£(EQ,STRING1,STRING1,£(DS,EQUAL-NULL,THIS EQUAL FUNCTION HAS NULL VALUE),ERROR)'  37
£(EQ,STRING,TEST,ERROR FUNCTION,(£(CL,EQUAL-NULL)))'                         38
```

```
£(EQ,STRING1,STRING2,ERROR,£(DS,EQUAL-NULL1,THIS EQ FUNCTION HAS NULL VALUE))'
£(EQ,STRING,STRING,(£(CL,EQUAL-NULL1)),ERROR)'
£(TF)'
£(DA)'
£(RS)£(RS)£(RS)£(RS)£(RS)£(RS)£(C£(PS,THE DEFINITION OF FORMS BEGINS)S,£(RS))
£(DS,,ABC IS MY ABDC) £(SS,,ABC,ABDC)          £(DS,£(RS))£(DS,£(RS))£(DS,£(RS
                                    ))££(SS,X2,MARKER1,MARKER2,MARKER3)££(SS,X4,PARI
S,LEARNING,FRENCH)££(SS,X1,SMITH,SEAT,HOUSE,COMMONS)£(RS)£(RS)£(RS)£(RS)£(RS)£(R
S)£(SS,X3,MARKER1,MARKER2,MARKER3,MARKER4)£(RS)£(RS)£(PS,£(CL,))£(PS,£(CL,,))£(P
S,£(CL,,THIS,BOOK))'
£(PS,                    STORE  OF  FORMS)'
£(PS,**************************************************************)'
£(PS,             X1  MR SMITH HAS A SEAT IN THE HOUSE OF COMMONS)'
£(PS,             X2  THE CLASSIC ROUTE TO THE MIDNIGHT SUN)'
£(PS,             X3  AFTER LUNCH YOU CONTINUE BY TRAIN A ROOM IS RESERVED FOR Y
OU)'
£(PS,             X4  HE WENT TO PARIS WITH THE INTENTION OF LEARNING FRENCH)'
X1,MR SMITH HAS A SEAT IN THE HOUSE OF COMMONS'
X2,MARKER1 CLASSIC MARKER2 TO MARKER1 MARKER3'
X3,AFTER LUNCH MARKER1 CONTINUE MARKER3 MARKER4 MARKER2 A ROOM IS RESERVED FOR M
ARKER1'
X4,HE WENT TO PARIS WITH THE INTENTION OF LEARNING FRENCH'
£(PS,£(CLX1,TAYLOR,FLAT,CITY,LONDON)£(CL,X1,TAYLOR,FLAT,CITY,LONDON))'
£(PS,£(CL,X2))'
£(PS,£(CL,X2,THE,ROUTE,MIDNIGHT SUN))'
£(PS,£(CL,X2,))'
£(PS,£(CL,X3,YOU,BY TRAIN))'
£(PS,£(CL,X5))'
£(PS,£(CL,X3,YOU,BY TRAIN))'
£(PS,£(CL,X4,STOCKHOLM,MEETING,FRIENDS,TELEIE,COSTA))'
£(TN)'
£(DS,EQUAL,(£(EQ,£(RS),X1,(£(PS,££(CL,X1))),(£(PS,£(CL,))))£(CL,EQUAL))£(CL,EQU
AL)'
X1'
FUNNY NAME'
£(PF,X1)£(PF,X2)£(PF,X3)£(PF,)£(LN,NAME  )'
£(LN,NAME)££(LN,NAME  )£(LN COMMA MISSING)£(LN,)'
££(TF)'
```

```
£(DA)'                                                                                                  77
£(PF,X1)£(PF,X2)£(PF,X3)£(PF,)£(LN,NAME  )'                                                             78
£(PS,(((CCSTAS)STATHCPCULCS)))'                                                                         79
£(PS,COSTAS££STATHCPCULCS)'                                                                             80
£(PS,COSTAS£STATHCPCULCS)'                                                                              81
£(PS,COSTAS£££(PS,STATHCPCULCS))'                                                                       82
£(ABC,DATA STRING*/-.....  ERRCR1)'                                                                     83
££(DEF,12+35=47   ERRCR2)'                                                                              84
£(PS,FIRST CHARACTER CF SECCND CARD IS QUCTE AND EXECUTION DCES NOT TERMINATE  )                        85
'                                                                                                       86
£(DA)'SECCND PART OF TEST                                                                               87
££(PS,-------------------------------------------------------------------------)                       88
££(DA)'                                                                                                 89
£(DS,TEXT-NAME,THERE IS NC SMCKE WITHCUT FIRE)'                                                         90
££(CL,TEXT-NAME)'                                                                                       91
£(DS,TEXT-NAME,THERE IS NC SMCKE WITHCUT FIRE)£(CL,TEXT-NAME)'                                          92
£(TN)'                                                                                                  93
£(DS,£(PS,GIVE NAME FCR TEXT)£(RS),£(PS,GIVE THE TEXT)£(RS))£(CL,£(PS,GIVE THE N                        94
AME CF THE TEXT)£(RS))'                                                                                 95
TEXT-NAME'                                                                                              96
THERE IS NO SMCKE WITHCUT FIRE'                                                                         97
TEXT-NAME'                                                                                              98
££(TF)'                                                                                                 99
£(DS,TEXT STCRING,(£(DS,£(PS,GIVE NAME FCR TEXT)£(RS),£(PS,GIVE THE TEXT)£(RS))£                       100
(CL,£(PS,GIVE THE NAME CF THE TEXT)£(RS))))'                                                           101
£(CL,TEXT STCRING)£(CL,TEXT STCRING)'                                                                  102
TEXT-NAME'                                                                                             103
THERE IS NO SMCKE WITHCUT FIRE'                                                                        104
TEXT-NAME'                                                                                             105
MY NAME'                                                                                               106
COSTAS NICKCLAS STATHCPCULCS'                                                                          107
MY NAME'                                                                                               108
£(TN)'                                                                                                 109
£(DS,EQLAL,(£(EQ,£(RS),TEXT-NAME,(£(PS,££(CL,TEXT-NAME))),(£(PS,££(CL,MY NAME)))                      110
)£(CL,EQLAL)))£(CL,EQUAL)'                                                                             111
TEXT-NAME'                                                                                             112
FUNNY-NAME'                                                                                            113
£(PF,TEXT-NAME)£(PF,MY NAME)£(PF,TEXT STCRING)£(LN,NAME  )'                                            114
```

```
£(DA)'                                                                          115
£(PF,TEXT-NAME)£(PF,MY NAME)£(PF,TEXT STORING)£(LN,NAME  )'                     116
££(PS,£(PS,CHARACTER STRING))'                                                  117
£(DS,,STRING1 WITH NULL NAME)£(PF,)££(STRING2 WITH NULL NAME)££(DS,,STRING2 WITH 118
 NULL NAME)'                                                                    119
£(CL,)'                                                                         120
£(DS STRING3,COMMA MISSING AFTER 1ST ARG.)                                      121
£(DS,STRING4 WITH NULL VALUE)                                                   122
£(DS,STRING5 WITH NULL VALUE,)'                                                 123
£(CL,STRING4 WITH NULL VALUE)'                                                  124
£(CL,STRING5 WITH NULL VALUE)'                                                  125
£(LN,NAME   )'                                                                  126
£(DS,CCBCL PRCGRAM,                                                             127
THIS ARTICLE CATEGCRISES VARICUS CCMPUTER TRAINING NEECS THAT EXIST TCDAY        128
AND CONSIDERS WHC IS BEST ABLE TC PROVIDE                                       129
SUITABLE CCURSES .SUGGESTICNS ARE                                               130
MADE AS TC WHAT ACTICN                                                          131
IS REQUIRED TC CC-CRDINATE                                                      132
)'                                                                              133
£(DA)'                                                                          134
£(DS,AFTER DELETICN,THIS FCRM IS CREATED AFTER DELETING ALL EXISTING FCRMS FRCM 135
FSTCRE)'                                                                        136
£(CL,AFTER DELETICN)'                                                           137
 C01030 AUTHOR. CCSTAS-ST'                                                      138
£(DS,RIGHT PARENTHESIS,££(RS))'                                                 139
)'                                                                              140
££(CL,RIGHT PARENTHESIS)'                                                       141
£(DS,LEFT PARENTHESIS,££(RS))'                                                  142
('                                                                              143
££(CL,LEFT PARENTHESIS)'                                                        144
£(DS,FCRM,THE PCINTS REQUIRED DESCRIBE THE SHAPE)'                              145
£(SS FCRM,A,B,C)'                                                               146
££(SS,FCRM)'                                                                    147
£(SS,FCRM,)'                                                                    148
£(SS,FCRM,,PCINTS,,RE,)'                                                        149
£(PF,FCRM)'                                                                     150
£(DS,FCRM,££(CL,FCRM,PCINTS,RE))'                                               151
£(SS,FCRM,PCINTS,RE)'                                                           152
```

```
£(PF,FORM)'                                                                    153
£(DS,REQUIEM,UNDER THE WIDE AND STARRY SKY/DIG THE GRAVE AND LET ME LIE./GLAD DI 154
D I LIVE AND GLADLY DIE,/AND I LAID ME DOWN WITH A WILL./ THIS BE THE VERSE YOU  155
GRAVE FOR ME/HERE HE LIES WHERE HE LONGED TO BE/HOME IS THE SAILOR HOME FROM SEA 156
 /AND THE HUNTER HOM FROM THE HILL.)'                                          157
£(SS,REQUIEM,GRAVE,LIES)'                                                      158
£(DA)'                                                                         159
£(DS,EMPTY FSTORE,FORM CREATED AFTER  SS FUNCTION MESSAGE)'                    160
£(CL,EMPTY FSTORE)'                                                            161
 ££(PS ALL IS FLUX NOTHING IS STATIONARY.)££(PS,ALL IS FLUX NOTHING IS STATIONAR 162
Y.)'                                                                           163
£(TN)'                                                                         164
£(DS,FORM1,THIS FORM WILL BE REDEFINED)£(DS,FORM2,THAN TO RUST OUT)'            165
£(CL,FORM1)/£(CL,FORM2)'                                                       166
£(DS,FORM1,IT IS BETTER TO WEAR OUT)'                                          167
£(CL,FORM1)/£(CL,FORM2)'                                                       168
£(DS,,)£(CL,)'                                                                 169
£(SS,ORM1,I,T)£(SS,FORM1,I,T)£(PF,FORM1)'                                      170
£(DS,FORM1,FORM1 IS REDEFINED DOES NOT CONTAIN MARKERS)'                       171
£(SS,FORM1,NON-EXISTING-1,NON-EXISTING-2,NON-EXISTING-3)'                      172
£(PF FORM2)£(PF,FORM2)'                                                        173
£(PF,UNDEFINED FORM)'                                                          174
£(PS ££(CL,FORM1))£(PS,££(CL,FORM1))'                                          175
£(DA)'                                                                         176
£(LN,STRING)'                                                                  177
£(DA)'                                                                         178
£(TN)'                                                                         179
£(EQ A2,A3,A4,A5)'                                                             180
£(EQ,A2,A3,A4)'                                                                181
£(EQ,A2,A3)'                                                                   182
£(EQ,A2)'                                                                      183
£(EQ)'                                                                         184
£(EQ,A,A,,ERROR)'                                                              185
£(EQ A2,A3,A4,A5)'                                                             186
£(EQ,A2,A3,A4)'                                                                187
£(EQ,A2,A2,VALUE FOURTH,ERROR)'                                                188
£(EQ,,,VALUE FOURTH,ERROR)'                                                    189
£(EQ,£(DS,FORM1,NOT NULL),£(DS,FORM2),VALUE FOURTH,£(DS,FORM3,))'              190
```

```
£(EQ,A2,A2,£(CL,FORM3),ERROR)'                                                  191
£(EQ,A2,A3,ERROR,VALUE FIFTH)'                                                  192
£(EQ,,A3,ERROR,VALUE FIFTH)'                                                    193
£(EQ,£(CL,FORM2),£(CL,FORM1),ERROR,VALUE FIFTH)'                                194
£(EQ,£(CL,FORM1),£(CL,FORM3),ERROR,£(CL,FORM2))'                                195
'                                                                               196
                    PROCESSOR  FULL FROM AN ACTIVE  RS FUNCTION                 197
£(TN)'                                                                          198
£(DS,PROCESSOR FULL FROM RS,£(RS))'                                             199
CARD1  RS                                                                       200
CARD  RS                                                                        201
        CARD3   RS                                                              202
CARD 4    RS                                                                    203
CARD 5   RS                                                                     204
    CARD   6   RS                                                               205
         CARD   7   RS                                                          206
     CARD  8   RS                                                               207
         CARD  9  RS                                                            208
        CARD  10   RS                                                           209
       CARD  11                                                                 210
      CARD   12                                                                 211
        CARD   13                                                               212
        CARD   14                                                               213
'                                                                               214
                    PROCESSOR  FULL FROM A  NEUTRAL RS  FUNCTION                215
£(EQ,STRING,STRING,(££(RS)),ERROR FUNCTION)'                                    216
     CARD1                                                                      217
     CARD2                                                                      218
              CARD 3                                                            219
             CARD4                                                              220
          CARDK  K                                                              221
CARD 6                                                                          222
CARD 7                                                                          223
      CARD 8                                                                    224
CARD 9                                                                          225
    CARD   10                                                                   226
      CARD   11                                                                 227
        CARD  12                                                                228
```

```
     CARD13                                                                  ' 229
'                                                                              230
                    PROCESSOR  FULL FROM AN ACTIVE  CL FUNCTION                231
£(DS,TEST FOR CL,                CARD1-CL                                      232
                     CARD2- CL                                                 233
               CARD3-CL CCL.28)'                                               234
£(PS,££(RS)£(CL,TEST FOR CL))'                                                 235
     CARD1                                                                     236
          CARD   2                                                             237
          CARD3                                                                238
            CARD4                                                              239
         CARD5                                                                 240
           CARD6                                                               241
                CARD7                                                          242
               CARD8                                                           243
               CARD9                                                           244
                  CARD10                                                       245
          CARD  11                                                           ' 246
'                                                                              247
                    PROCESSOR  FULL FROM A  NEUTRAL CL  FUNCTION               248
£(DS,TEST-CL,          CARD1                                                   249
                     CARD2                                                     250
                CARD3                                                          251
                  CARD4                                                        252
               CARD5                                                         )' 253
££(CL,TEST-CL)££(CL,TEST-CL)££(CL,TEST-CL)££(CL,TEST-CL)'                      254
'                                                                              255
                    GARBAGE   COLLECTION                                       256
£(TN)'                                                                         257
£(PS,TEST FOR ORDINARY CHARACTER WHEN TRAC PROCESSOR FULL)                     258
             CARD1                                                             259
             CARD2                                                             260
             CARD3                                                             261
             CARD4                                                             262
             CARD5                                                             263
             CARD6                                                             264
             CARD7                                                             265
             CARD8                                                             266
```

CARD9 267
CARD10 268
CARD13 269
270

IIB.14 Results

. . . pages 121-151

```
                                    RESULTS                                                                    1
*******************************************************************************************************      2
 +++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                                                 3
 PS"                                                                                                           4
 +++++THIS PS FUNCTICN HAS NULL STRING+++++                                                                    5
 +++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                                                 6
 RS                                                                                                            7
 DS"NULL VALUE FCRM1"                                                                                          8
 +++++THIS STRING HAS NULL VALUE+++++                                                                          9
 +++++DS"NULL VALUE FCRM1"                                                                                    10
 DS"NULL VALUE FCRM2                                                                                          11
 +++++THIS STRING HAS NULL VALUE+++++                                                                         12
 +++++DS"NULL VALUE FCRM2                                                                                     13
 PF"NULL VALUE FCRM1                                                                                          14
 +++++NULL VALUE FCRM+++++                                                                                    15
 +++++PF"NULL VALUE FCRM1                                                                                     16
 PF"NULL VALUE FCRM2                                                                                          17
 +++++NULL VALUE FCRM+++++                                                                                    18
 +++++PF"NULL VALUE FCRM2                                                                                     19
 PS"                                                                                                          20
 +++++THIS PS FUNCTICN HAS NULL STRING+++++                                                                   21
 +++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                                                22
 RS                                                                                                           23
 DS"NULL VALUE FCRM1"FCRM1 IS REDEFINED IT IS NOT NULL ANY MCRE                                               24
 PF"NULL VALUE FCRM1                                                                                          25
 FCRM1 IS REDEFINED IT IS NCT NULL ANY MCRE                                                                   26
 PS"                                                                                                          27
                                                                                                              28
 +++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                                                29
 RS                                                                                                           30
 DA                                                                                                           31
 PS"                                                                                                          32
 +++++THIS PS FUNCTICN HAS NULL STRING+++++                                                                   33
 +++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                                                34
 RS                                                                                                           35
 DS"FCRM1"    FIRST DEFINED FCRM                                                                              36
 DS"FCRM2"2ND DEFINED A FCRM B     C                                                                          37
 DS"FCRM3"3RD DEFINED FCRM                                                                                    38
```

```
SS"FORM2"A"B"C                                                    39
PS"                                                               40
+++++THIS PS FUNCTION HAS NULL STRING+++++                        41
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                     42
RS                                                                43
DS"B"BBB                                                          44
PS"B                                                              45
B                                                                 46
CL"FORM2"AAA""BBB""CCC                                            47
PS"2ND DEFINED AAA FORM BBB     CCC                               48
2ND DEFINED AAA FORM BBB     CCC                                  49
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                     50
RS                                                                51
DA                                                                52
PS"                                                               53
+++++THIS PS FUNCTION HAS NULL STRING+++++                        54
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                     55
RS                                                                56
DS"STRING1 WITH NULL VALUE"                                       57
+++++THIS STRING HAS NULL VALUE+++++                              58
+++++DS"STRING1 WITH NULL VALUE"                                  59
PS"                                                               60
+++++THIS PS FUNCTION HAS NULL STRING+++++                        61
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                     62
RS                                                                63
DS"STRING2 WITH NULL VALUE                                        64
+++++THIS STRING HAS NULL VALUE+++++                              65
+++++DS"STRING2 WITH NULL VALUE                                   66
PS"                                                               67
+++++THIS PS FUNCTION HAS NULL STRING+++++                        68
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                     69
RS                                                                70
CL"STRING1 WITH NULL VALUE                                        71
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                 72
+++++CALL TO A FORM WITH NULL VALUE+++++                          73
+++++CL"STRING1 WITH NULL VALUE                                   74
PS"                                                               75
+++++THIS PS FUNCTION HAS NULL STRING+++++                        76
```

```
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                          77
RS                                                                     78
CL"STRING2 WITH NULL VALUE                                             79
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                       80
+++++CALL TC A FCRM WITH NULL VALUE+++++                                81
+++++CL"STRING2 WITH NULL VALUE                                        82
PS"                                                                    83
+++++THIS PS FUNCTICN HAS NULL STRING+++++                              84
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                          85
RS                                                                     86
CL"STRING1 WITH NULL VALUE                                             87
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                       88
+++++CALL TC A FCRM WITH NULL VALUE+++++                                89
+++++CL"STRING1 WITH NULL VALUE                                        90
CL"STRING2 WITH NULL VALUE                                             91
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                       92
+++++CALL TC A FCRM WITH NULL VALUE+++++                                93
+++++CL"STRING2 WITH NULL VALUE                                        94
EQ"""STRINGS EQUAL SINCE BCTH HAVE NULL VALUE"ERRCR                    95
PS"STRINGS EQUAL SINCE BCTH HAVE NULL VALUE                            96
STRINGS EQUAL SINCE BCTH HAVE NULL VALUE                               97
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                          98
RS                                                                     99
DA                                                                    100
PS"                                                                   101
+++++THIS PS FUNCTICN HAS NULL STRING+++++                            102
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                         103
RS                                                                    104
DS"FCRM1"    FIRST DEFINED FCRM                                       105
DS"FCRM2"2ND DEFINED A FCRM B     C                                   106
DS"FCRM3"3RD DEFINED FCRM                                             107
SS"FCRM2"A"B"C                                                        108
PS"                                                                   109
+++++THIS PS FUNCTICN HAS NULL STRING+++++                            110
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                         111
RS                                                                    112
CL"FCRM2"   CARD1                                                     113
         CARD2                                                    "   114
```

```
            CARD3                                                                    115
                  CARD4                                                              116
"                  CARD5                                                            117
          CARD6                                                                     118
                                                                                    119
+++++NC PLACE IN FSTORE FORM CANNOT BE CALLED+++++                                  120
+++++CL"FORM2"   CARD1                                                  +++++      121
+++++                  CARD2                                            +++++      122
+++++                  "          CARD3                                 +++++      123
+++++                                        CARD4                     +++++      124
+++++                                              "          CARD5     +++++      125
+++++                                                          CARD6     +++++      126
+++++                                                                               127
PS"                                                                                 128
+++++THIS PS FUNCTION HAS NULL STRING+++++                                          129
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                       130
RS                                                                                  131
DA                                                                                  132
PS"                                                                                 133
+++++THIS PS FUNCTION HAS NULL STRING+++++                                          134
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                       135
RS                                                                                  136
DS"TESTEQ"2ND AND 3RD ARGUMENTS OF EQ FUNCTION ARE EQUAL SINCE BOTH NULL            137
PS"                                                                                 138
+++++THIS PS FUNCTION HAS NULL STRING+++++                                          139
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                       140
RS                                                                                  141
EQ"""£(CL,TESTEQ)"ERROR FUNCTION                                                    142
CL"TESTEQ                                                                           143
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                   144
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                    145
PS"2ND AND 3RD ARGUMENTS OF EQ FUNCTION ARE EQUAL SINCE BOTH NULL                   146
2ND AND 3RD ARGUMENTS OF EQ FUNCTION ARE EQUAL SINCE BOTH NULL                      147
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                       148
RS                                                                                  149
DS"EQUAL-NULL"THIS EQUAL FUNCTION HAS NULL VALUE                                    150
EQ"STRING1"STRING1""ERROR                                                           151
PS"                                                                                 152
```

```
+++++THIS PS FUNCTION HAS NULL STRING+++++                                153
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                             154
RS                                                                       155
EQ"STRING"TEST"ERROR FUNCTION"£(CL,EQUAL-NULL)                           156
CL"EQUAL-NULL                                                            157
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                         158
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++          159
PS"THIS EQUAL FUNCTION HAS NULL VALUE                                    160
THIS EQUAL FUNCTION HAS NULL VALUE                                       161
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                             162
RS                                                                       163
DS"EQUAL-NULL1"THIS EQ FUNCTION HAS NULL VALUE                           164
EQ"STRING1"STRING2"ERROR"                                                165
PS"                                                                      166
+++++THIS PS FUNCTION HAS NULL STRING+++++                                167
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                             168
RS                                                                       169
EQ"STRING"STRING"£(CL,EQUAL-NULL1)"ERROR                                 170
CL"EQUAL-NULL1                                                           171
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                         172
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++          173
PS"THIS EQ FUNCTION HAS NULL VALUE                                       174
THIS EQ FUNCTION HAS NULL VALUE                                          175
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                             176
RS                                                                       177
TF                                                                       178
+++++THIS PS FUNCTION HAS NULL STRING+++++                                179
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                             180
+++++THIS PS FUNCTION HAS NULL STRING+++++                                181
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                             182
                    STORE  OF  FORMS                                     183
************************************************************************ 184
            X1  MR SMITH HAS A SEAT IN THE HOUSE OF COMMONS              185
            X2  THE CLASSIC ROUTE TO THE MIDNIGHT SUN                    186
            X3  AFTER LUNCH YOU CONTINUE BY TRAIN A ROOM IS RESERVED FOR YOU 187
            X4  HE WENT TO PARIS WITH THE INTENTION OF LEARNING FRENCH    188
THE DEFINITION OF FORMS BEGINS                                           189
+++++ERROR FUNCTION ,COMMA IS MISSING AFTER FIRST ARGUMENT+++++          190
```

```
+++++CLX1"TAYLOR"FLAT"CITY"LONDON                                              191
MR TAYLOR HAS A FLAT IN THE CITY OF LONDON                                     192
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              193
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                 194
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                 195
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                 196
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                 197
 CLASSIC  TO                                                                   198
THE CLASSIC ROUTE TO THE MIDNIGHT SUN                                          199
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              200
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                 201
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                 202
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                 203
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                 204
 CLASSIC  TO                                                                   205
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++               206
AFTER LUNCH MARKER1 CONTINUE MARKER3 MARKER4 MARKER2 A ROOM IS RESERVED FOR MAR 207
KER1                                                                           208
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              209
+++++THERE IS NO SUCH FORM IN FSTORE TO BE CALLED   +++++                      210
+++++CL"X5                                                                     211
+++++THIS PS FUNCTION HAS NULL STRING+++++                                     212
+++++NUMBER OF ARGUMENTS LESS THAN NO OF MARKERS                               213
NOMARKERS-NOARG MARKERS REPLACED BY NULL+++++                                  214
+++++NUMBER OF ARGUMENTS LESS THAN NO OF MARKERS                               215
NOMARKERS-NOARG MARKERS REPLACED BY NULL+++++                                  216
AFTER LUNCH YOU CONTINUE   BY TRAIN A ROOM IS RESERVED FOR YOU                 217
HE WENT TO STOCKHOLM WITH THE INTENTION OF MEETING FRIENDS                     218
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              219
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                 220
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                 221
 IS MY                                                                         222
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              223
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                 224
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                 225
 IS MY                                                                         226
THIS IS MY BOOK                                                                227
                                                                               228
```

```
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++
PS"
+++++THIS PS FUNCTICN HAS NULL STRING+++++
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++
RS
DS"EQUAL"£(EQ,£(RS),X1,(£(PS,££(CL,X1))),(£(PS,£(CL,))))£(CL,EQUAL)
CL"EQUAL
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++
+++++FCRM WITH NC MARKERS ,ARGUMENTS OF CL FUNCTICN IGNORED+++++
RS
EQ"X1"X1"£(PS,££(CL,X1))"£(PS,£(CL,))
CL"X1
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++
+++++THE CL FUNC HAS NC ARGUMENTS MARKERS REPLACED BY NULL+++++
+++++THE CL FUNC HAS NC ARGUMENTS MARKERS REPLACED BY NULL+++++
+++++THE CL FUNC HAS NC ARGUMENTS MARKERS REPLACED BY NULL+++++
+++++THE CL FUNC HAS NC ARGUMENTS MARKERS REPLACED BY NULL+++++
PS"MR  HAS A  IN THE  CF
MR  HAS A  IN THE  CF
CL"EQUAL
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++
+++++FCRM WITH NC MARKERS ,ARGUMENTS OF CL FUNCTICN IGNORED+++++
RS
EQ"FUNNY NAME"X1"£(PS,££(CL,X1))"£(PS,£(CL,))
CL"
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++
+++++THE CL FUNC HAS NC ARGUMENTS MARKERS REPLACED BY NULL+++++
+++++THE CL FUNC HAS NC ARGUMENTS MARKERS REPLACED BY NULL+++++
PS" IS MY
 IS MY
CL"EQUAL
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++
+++++FCRM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTICN IGNORED+++++
RS
PF"X1
MR &1 HAS A &2 IN THE &3 CF &4
PF"X2
&1 CLASSIC &2 TC &1 &3
```

```
PF"X3                                                                    267
AFTER LUNCH &1 CONTINUE &3 &4 &2 A ROOM IS RESERVED FOR &1               268
PF"                                                                      269
&1 IS MY &2                                                              270
LN"NAME                                                                  271
NAME                                                                     272
NAME  X2                                                                 273
NAME  X4                                                                 274
NAME  X1                                                                 275
NAME  X3                                                                 276
NAME  EQUAL                                                              277
+++++THERE ARE NO FORMS IN STORE+++++                                    278
EQ""X1"&(PS,&&(CL,X1))"&(PS,&(CL,))                                     279
CL"                                                                      280
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                        281
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++          282
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++          283
PS" IS MY                                                                284
 IS MY                                                                   285
CL"EQUAL                                                                 286
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                        287
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++         288
RS                                                                       289
LN"NAME                                                                  290
NAME                                                                     291
NAMEX2                                                                   292
NAMEX4                                                                   293
NAMEX1                                                                   294
NAMEX3                                                                   295
NAMEEQUAL                                                                296
+++++THERE ARE NO FORMS IN STORE+++++                                    297
LN"NAME                                                                  298
NAME                                                                     299
NAME  X2                                                                 300
NAME  X4                                                                 301
NAME  X1                                                                 302
NAME  X3                                                                 303
NAME  EQUAL                                                              304
```

```
+++++THERE ARE NC FCRMS IN STCRE+++++                                  305
LN COMMA MISSING                                                       306
+++++ERROR FUNCTICN,CCMMA IS MISSING AFTER FIRST ARGUMENT+++++         307
+++++LN COMMA MISSING                                                  308
LN"                                                                    309
X2                                                                     310
X4                                                                     311
X1                                                                     312
X3                                                                     313
EQUAL                                                                  314
+++++THERE ARE NC FCRMS IN STCRE+++++                                  315
EQ""X1"£(PS,££(CL,X1))"£(PS,£(CL,))                                   316
CL"                                                                    317
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                      318
+++++THE CL FUNC HAS NC ARGUMENTS MARKERS REPLACED BY NULL+++++        319
+++++THE CL FUNC HAS NC ARGUMENTS MARKERS REPLACED BY NULL+++++        320
PS" IS MY                                                              321
 IS MY                                                                 322
CL"EQUAL                                                               323
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                      324
+++++FCRM WITH NC MARKERS ,ARGUMENTS OF CL FUNCTICN IGNCRED+++++       325
RS                                                                     326
TF                                                                     327
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                      328
+++++THE CL FUNC HAS NC ARGUMENTS MARKERS REPLACED BY NULL+++++        329
+++++THE CL FUNC HAS NC ARGUMENTS MARKERS REPLACED BY NULL+++++        330
 IS MY                                                                 331
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                      332
+++++FCRM WITH NC MARKERS ,ARGUMENTS OF CL FUNCTICN IGNCRED+++++       333
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTICN+++++                      334
+++++THERE IS NC SUCH FCRM IN FSTCRE TO BE CALLED   +++++             335
+++++CL"                                                               336
+++++THIS PS FUNCTICN HAS NULL STRING+++++                            337
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                      338
+++++THERE IS NO SUCH FCRM IN FSTCRE TO BE CALLED   +++++             339
+++++CL"EQUAL                                                          340
+++++THIS PS FUNCTICN HAS NULL STRING+++++                            341
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                         342
```

```
+++++ THERE IS NCT SUCH A FCRM IN STCRE+++++
+++++PF"X1
+++++ THERE IS NCT SUCH A FCRM IN STCRE+++++
+++++PF"X2
+++++ THERE IS NCT SUCH A FCRM IN STCRE+++++
+++++PF"X3
+++++ THERE IS NCT SUCH A FCRM IN STCRE+++++
+++++PF"
+++++THERE ARE NC FCRMS IN STCRE+++++
+++++THIS PS FUNCTICN HAS NULL STRING+++++
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++
((COSTAS)STATHCPCULCS)
+++++THIS PS FUNCTICN HAS NULL STRING+++++
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++
COSTAS$$STATHCPCULCS
+++++THIS PS FUNCTICN HAS NULL STRING+++++
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++
COSTAS$STATHCPCULCS
+++++THIS PS FUNCTICN HAS NULL STRING+++++
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++
STATHCPCULCS
COSTAS$
+++++THIS PS FUNCTICN HAS NULL STRING+++++
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++
+++++ERROR CODED FUNCTION DELETED+++++
+++++ABC"DATA STRING*/-.....  ERRCR1
+++++THIS PS FUNCTICN HAS NULL STRING+++++
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++
+++++ERROR CODED FUNCTION DELETED+++++
+++++DEF"12+35=47   ERROR2
+++++THIS PS FUNCTICN HAS NULL STRING+++++
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++
FIRST CHARACTER OF SECCND CARD IS QUCTE ANC EXECUTICN DOES NCT TERMINATE
+++++THIS PS FUNCTICN HAS NULL STRING+++++
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++
+++++THIS PS FUNCTICN HAS NULL STRING+++++
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++
------------------------------------------------------------------
```

```
+++++THIS PS FUNCTION HAS NULL STRING+++++                                  381
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                               382
+++++THIS PS FUNCTION HAS NULL STRING+++++                                  383
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                               384
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                           385
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++            386
THERE IS NO SMOKE WITHOUT FIRE                                             387
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                               388
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                           389
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++            390
THERE IS NO SMOKE WITHOUT FIRE                                             391
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                               392
PS"                                                                        393
+++++THIS PS FUNCTION HAS NULL STRING+++++                                  394
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                               395
RS                                                                         396
PS"GIVE NAME FOR TEXT                                                      397
GIVE NAME FOR TEXT                                                         398
RS                                                                         399
PS"GIVE THE TEXT                                                           400
GIVE THE TEXT                                                              401
RS                                                                         402
DS"TEXT-NAME"THERE IS NO SMOKE WITHOUT FIRE                                403
PS"GIVE THE NAME OF THE TEXT                                               404
GIVE THE NAME OF THE TEXT                                                  405
RS                                                                         406
CL"TEXT-NAME                                                               407
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                          408
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++           409
PS"THERE IS NO SMOKE WITHOUT FIRE                                          410
THERE IS NO SMOKE WITHOUT FIRE                                             411
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                              412
RS                                                                         413
TF                                                                         414
+++++THIS PS FUNCTION HAS NULL STRING+++++                                 415
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                              416
+++++THIS PS FUNCTION HAS NULL STRING+++++                                 417
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                              418
```

```
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                419
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 420
GIVE NAME FOR TEXT                                                              421
GIVE THE TEXT                                                                   422
GIVE THE NAME OF THE TEXT                                                       423
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                424
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 425
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                426
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 427
GIVE NAME FOR TEXT                                                              428
GIVE THE TEXT                                                                   429
GIVE THE NAME OF THE TEXT                                                       430
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                431
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 432
THERE IS NO SMOKE WITHOUT FIRE00STAS NICK0LAS STATHOPOULOS                       433
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                    434
PS"                                                                             435
+++++THIS PS FUNCTION HAS NULL STRING+++++                                       436
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                    437
RS                                                                              438
DS"EQUAL"£(EQ,£(RS),TEXT-NAME,(£(PS,££(CL,TEXT-NAME))),(£(PS,££(CL,MY NAME))))£ 439
(CL,EQUAL)                                                                      440
CL"EQUAL                                                                        441
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                442
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 443
RS                                                                              444
EQ"TEXT-NAME"TEXT-NAME"£(PS,££(CL,TEXT-NAME))"£(PS,££(CL,MY NAME))              445
CL"TEXT-NAME                                                                    446
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                447
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 448
PS"THERE IS NO SMOKE WITHOUT FIRE                                               449
THERE IS NO SMOKE WITHOUT FIRE                                                  450
CL"EQUAL                                                                        451
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                452
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 453
RS                                                                              454
EQ"FUNNY-NAME"TEXT-NAME"£(PS,££(CL,TEXT-NAME))"£(PS,££(CL,MY NAME))             455
CL"MY NAME                                                                      456
```

```
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              457
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++               458
PS"COSTAS NICKOLAS STATHOPOULOS                                               459
COSTAS NICKOLAS STATHOPOULOS                                                  460
CL"EQUAL                                                                      461
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              462
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++               463
RS                                                                            464
PF"TEXT-NAME                                                                  465
THERE IS NO SMOKE WITHOUT FIRE                                                466
PF"MY NAME                                                                    467
COSTAS NICKOLAS STATHOPOULOS                                                  468
PF"TEXT STORING                                                               469
£(DS,£(PS,GIVE NAME FOR TEXT)£(RS),£(PS,GIVE THE TEXT)£(RS))£(CL,£(PS,GIVE THE   470
NAME OF THE TEXT)£(RS))                                                       471
LN"NAME                                                                       472
NAME  TEXT STORING                                                            473
NAME  TEXT-NAME                                                               474
NAME  MY NAME                                                                 475
NAME  EQUAL                                                                   476
+++++THERE ARE NO FORMS IN STORE+++++                                         477
EQ""TEXT-NAME"£(PS,££(CL,TEXT-NAME))"£(PS,££(CL,MY NAME))                     478
CL"MY NAME                                                                    479
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              480
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++               481
PS"COSTAS NICKOLAS STATHOPOULOS                                               482
COSTAS NICKOLAS STATHOPOULOS                                                  483
CL"EQUAL                                                                      484
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              485
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++               486
RS                                                                            487
DA                                                                            488
EQ""TEXT-NAME"£(PS,££(CL,TEXT-NAME))"£(PS,££(CL,MY NAME))                     489
CL"MY NAME                                                                    490
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              491
+++++THERE IS NO SUCH FORM IN FSTORE TO BE CALLED   +++++                      492
+++++CL"MY NAME                                                               493
PS"                                                                           494
```

```
+++++THIS PS FUNCTION HAS NULL STRING+++++
CL"EQUAL
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++THERE IS NO SUCH FORM IN FSTORE TO BE CALLED   +++++
+++++CL"EQUAL
PS"
+++++THIS PS FUNCTION HAS NULL STRING+++++
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++
RS
PF"TEXT-NAME
+++++ THERE IS NOT SUCH A FORM IN STORE+++++
+++++PF"TEXT-NAME
PF"MY NAME
+++++ THERE IS NOT SUCH A FORM IN STORE+++++
+++++PF"MY NAME
PF"TEXT STORING
+++++ THERE IS NOT SUCH A FORM IN STORE+++++
+++++PF"TEXT STORING
LN"NAME
+++++THERE ARE NO FORMS IN STORE+++++
PS"
+++++THIS PS FUNCTION HAS NULL STRING+++++
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++
RS
PS"CHARACTER STRING
CHARACTER STRING
PS"
+++++THIS PS FUNCTION HAS NULL STRING+++++
PS"
+++++THIS PS FUNCTION HAS NULL STRING+++++
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++
RS
DS""STRING1 WITH NULL NAME
PF"
STRING1 WITH NULL NAME
STRING2 WITH NULL NAME
+++++ERROR CODED FUNCTION DELETED+++++
+++++STRING2 WITH NULL NAME
```

```
DS""STRING2 WITH NULL NAME                                                    533
PS"                                                                           534
+++++THIS PS FUNCTION HAS NULL STRING+++++                                    535
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                 536
RS                                                                            537
CL"                                                                           538
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                             539
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++              540
PS"STRING2 WITH NULL NAME                                                     541
STRING2 WITH NULL NAME                                                        542
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                 543
RS                                                                            544
DS STRING3"COMMA MISSING AFTER 1ST ARG.                                       545
+++++ERROR FUNCTION  THERE IS NO COMMA AFTER FIRST ARGUMENT+++++              546
+++++UNDEFINED STRING FOR THE FOLLOWING FUNCTION*****                         547
+++++DS STRING3"COMMA MISSING AFTER 1ST ARG.                                  548
DS"STRING4 WITH NULL VALUE                                                    549
+++++THIS STRING HAS NULL VALUE+++++                                          550
+++++DS"STRING4 WITH NULL VALUE                                               551
DS"STRING5 WITH NULL VALUE"                                                   552
+++++THIS STRING HAS NULL VALUE+++++                                          553
+++++DS"STRING5 WITH NULL VALUE"                                              554
PS"                                                                           555
                                                                              556
                                                                              557
                                                                              558
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                 559
RS                                                                            560
CL"STRING4 WITH NULL VALUE                                                    561
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                             562
+++++CALL TO A FORM WITH NULL VALUE+++++                                      563
+++++CL"STRING4 WITH NULL VALUE                                               564
PS"                                                                           565
+++++THIS PS FUNCTION HAS NULL STRING+++++                                    566
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                 567
RS                                                                            568
CL"STRING5 WITH NULL VALUE                                                    569
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                             570
```

```
+++++CALL TC A FCRM WITH NULL VALUE+++++                                    571
+++++CL"STRING5 WITH NULL VALUE                                             572
PS"                                                                         573
+++++THIS PS FUNCTICN HAS NULL STRING+++++                                  574
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                               575
RS                                                                          576
LN"NAME                                                                     577
NAME                                                                        578
NAME    STRING4 WITH NULL VALUE                                             579
NAME    STRING5 WITH NULL VALUE                                             580
+++++THERE ARE NC FCRMS IN STCRE+++++                                       581
PS"                                                                         582
+++++THIS PS FUNCTICN HAS NULL STRING+++++                                  583
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                               584
RS                                                                          585
DS"CCBCL PRCGRAM"                                                       T   586
HIS ARTICLE CATEGCRISES VARICUS CCMPUTER TRAINING NEEDS THAT EXIST TCDAY    587
AND CONSIDERS WHC IS BEST ABLE TC PRCVICE                                   588
 SUITABLE COURSES .SUGGESTICNS ARE                                          589
  MADE AS TC WHAT ACTICN                                                    590
   IS REQUIRED TC CC-CRDINATE                                               591
                                                                            592
+++++THERE IS NC PLACE IN STCRE AVAILABLE FCR FCRMS+++++                     593
+++++DS"CCBCL PRCGRAM"                                                 +++++ 594
+++++THERE IS NC PLACE IN STCRE AVAILABLE FCR FCRMS+++++                     595
+++++          THIS ARTICLE CATEGCRISES VARICUS CCMPUTER TRAINING NEEDS THA+++++ 596
+++++THERE IS NC PLACE IN STCRE AVAILABLE FCR FCRMS+++++                     597
+++++T EXIST TCDAY       AND CCNSIDERS WHC IS BEST ABLE TC PRCVICE       +++++ 598
+++++THERE IS NC PLACE IN STCRE AVAILABLE FCR FCRMS+++++                     599
+++++                                SUITABLE COURSES .SUGGESTICNS ARE  +++++ 600
+++++THERE IS NO PLACE IN STCRE AVAILABLE FCR FCRMS+++++                     601
+++++                                          MADE AS TC WHAT ACTICN   +++++ 602
+++++THERE IS NO PLACE IN STCRE AVAILABLE FCR FCRMS+++++                     603
+++++                                                IS REQUIRED TC C+++++ 604
+++++THERE IS NC PLACE IN STCRE AVAILABLE FCR FCRMS+++++                     605
+++++C-CRDINATE                                                            606
PS"                                                                         607
+++++THIS PS FUNCTICN HAS NULL STRING+++++                                  608
```

```
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                609
RS                                                                           610
DA                                                                           611
PS"                                                                          612
+++++THIS PS FUNCTICN HAS NULL STRING+++++                                   613
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                614
RS                                                                           615
DS"AFTER DELETICN"THIS FCRM IS CREATED AFTER DELETING ALL EXISTING FCRMS FRCM F  616
STORE                                                                        617
PS"                                                                          618
+++++THIS PS FUNCTICN HAS NULL STRING+++++                                   619
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                620
RS                                                                           621
CL"AFTER DELETICN                                                            622
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                            623
+++++FCRM WITH NC MARKERS ,ARGUMENTS CF CL FUNCTICN IGNCRED+++++             624
PS"THIS FORM IS CREATED AFTER DELETING ALL EXISTING FORMS FROM FSTORE        625
THIS FCRM IS CREATED AFTER DELETING ALL EXISTING FCRMS FRCM FSTORE            626
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                627
RS                                                                           628
PS" CC1C30 AUTHCR. CCSTAS-ST                                                 629
 CC103C AUTHCR. CCSTAS-ST                                                    630
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                631
RS                                                                           632
RS                                                                           633
DS"RIGHT PARENTHESIS")                                                       634
PS"                                                                          635
+++++THIS PS FUNCTICN HAS NULL STRING+++++                                   636
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                637
RS                                                                           638
CL"RIGHT PARENTHESIS                                                         639
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                            640
+++++FCRM WITH NC MARKERS ,ARGUMENTS CF CL FUNCTICN IGNCRED+++++             641
PS")                                                                         642
)                                                                            643
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                644
RS                                                                           645
RS                                                                           646
```

```
DS"LEFT PARENTHESIS"(                                                    647
PS"                                                                      648
+++++THIS PS FUNCTION HAS NULL STRING+++++                               649
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                            650
RS                                                                       651
CL"LEFT PARENTHESIS                                                      652
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                        653
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++         654
PS"(                                                                     655
(                                                                        656
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                            657
RS                                                                       658
DS"FORM"THE POINTS REQUIRED DESCRIBE THE SHAPE                           659
PS"                                                                      660
+++++THIS PS FUNCTION HAS NULL STRING+++++                               661
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                            662
RS                                                                       663
SS FORM"A"B"C                                                            664
+++++ERROR FUNCTION ,COMMA IS MISSING AFTER FIRST ARGUMENT+++++          665
+++++STRING NON SEGMENTED+++++                                           666
+++++SS FORM"A"B"C                                                       667
PS"                                                                      668
+++++THIS PS FUNCTION HAS NULL STRING+++++                               669
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                            670
RS                                                                       671
SS"FORM                                                                  672
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                        673
+++++SS"FORM                                                             674
PS"                                                                      675
+++++THIS PS FUNCTION HAS NULL STRING+++++                               676
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                            677
RS                                                                       678
SS"FORM"                                                                 679
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                        680
+++++SS"FORM"                                                            681
PS"                                                                      682
+++++THIS PS FUNCTION HAS NULL STRING+++++                               683
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                            684
```

```
RS                                                                                685
SS"FORM""POINTS""RE"                                                              686
PS"                                                                               687
+++++THIS PS FUNCTION HAS NULL STRING+++++                                        688
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                     689
RS                                                                                690
PF"FORM                                                                           691
THE &1 &2QUI&2D DESCRIBE THE SHAPE                                                692
PS"                                                                               693
+++++THIS PS FUNCTION HAS NULL STRING+++++                                        694
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                     695
RS                                                                                696
CL"FORM"POINTS"RE                                                                 697
DS"FORM"THE POINTS REQUIRED DESCRIBE THE SHAPE                                    698
PS"                                                                               699
+++++THIS PS FUNCTION HAS NULL STRING+++++                                        700
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                     701
RS                                                                                702
SS"FORM"POINTS"RE                                                                 703
PS"                                                                               704
+++++THIS PS FUNCTION HAS NULL STRING+++++                                        705
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                     706
RS                                                                                707
PF"FORM                                                                           708
THE &1 &2QUI&2D DESCRIBE THE SHAPE                                                709
PS"                                                                               710
+++++THIS PS FUNCTION HAS NULL STRING+++++                                        711
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                     712
RS                                                                                713
DS"REQUIEM"UNDER THE WIDE AND STARRY SKY/DIG THE GRAVE AND LET ME LIE./GLAD DID   714
 I LIVE AND GLADLY DIE"/AND I LAID ME DOWN WITH A WILL./ THIS BE THE VERSE YOU    715
GRAVE FOR ME/HERE HE LIES WHERE HE LONGED TO BE/HOME IS THE SAILOR HOME FROM SE   716
A /AND THE HUNTER HOM FROM THE HILL.                                              717
PS"                                                                               718
+++++THIS PS FUNCTION HAS NULL STRING+++++                                        719
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                     720
RS                                                                                721
SS"REQUIEM"GRAVE"LIES                                                             722
```

```
+++++FORM CANNOT BE SEGMENTED,STORE FULL+++++                          723
+++++SS"REQUIEM"GRAVE"LIES                                            724
PS"                                                                    725
+++++THIS PS FUNCTION HAS NULL STRING+++++                             726
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                          727
RS                                                                     728
DA                                                                     729
PS"                                                                    730
+++++THIS PS FUNCTION HAS NULL STRING+++++                             731
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                          732
RS                                                                     733
DS"EMPTY FSTORE"FORM CREATED AFTER  SS FUNCTION MESSAGE                734
PS"                                                                    735
+++++THIS PS FUNCTION HAS NULL STRING+++++                             736
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                          737
RS                                                                     738
CL"EMPTY FSTORE                                                        739
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                      740
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++       741
PS"FORM CREATED AFTER  SS FUNCTION MESSAGE                             742
FORM CREATED AFTER  SS FUNCTION MESSAGE                                743
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                          744
RS                                                                     745
PS ALL IS FLUX NOTHING IS STATIONARY.                                  746
+++++ERROR FUNCTION*****PS ALL IS FLUX NOTHING IS STATIONARY.          747
PS"ALL IS FLUX NOTHING IS STATIONARY.                                  748
ALL IS FLUX NOTHING IS STATIONARY.                                     749
PS"                                                                    750
                                                                       751
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                          752
RS                                                                     753
TN                                                                     754
PS"                                                                    755
+++++THIS PS FUNCTION HAS NULL STRING+++++                             756
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                          757
RS                                                                     758
DS"FORM1"THIS FORM WILL BE REDEFINED                                   759
DS"FORM2"THAN TO RUST OUT                                              760
```

```
PS"                                                                  761
+++++THIS PS FUNCTION HAS NULL STRING+++++                           762
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                        763
RS                                                                   764
CL"FORM1                                                             765
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    766
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     767
CL"FORM2                                                             768
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    769
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     770
PS"THIS FORM WILL BE REDEFINED/THAN TO RUST OUT                      771
THIS FORM WILL BE REDEFINED/THAN TO RUST OUT                         772
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                        773
RS                                                                   774
DS"FORM1"IT IS BETTER TO WEAR OUT                                    775
PS"                                                                  776
+++++THIS PS FUNCTION HAS NULL STRING+++++                           777
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                        778
RS                                                                   779
CL"FORM1                                                             780
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    781
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     782
CL"FORM2                                                             783
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    784
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     785
PS"IT IS BETTER TO WEAR OUT/THAN TO RUST OUT                         786
IT IS BETTER TO WEAR OUT/THAN TO RUST OUT                            787
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                        788
RS                                                                   789
DS""                                                                 790
+++++THIS STRING HAS NULL VALUE+++++                                 791
+++++DS""                                                            792
CL"                                                                  793
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    794
+++++CALL TO A FORM WITH NULL VALUE+++++                             795
+++++CL"                                                             796
PS"                                                                  797
+++++THIS PS FUNCTION HAS NULL STRING+++++                           798
```

```
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                          799
RS                                                                     800
SS"ORM1"I"T                                                            801
+++++THERE IS NC SUCH FCRM IN FSTCRE TC BE SEGMENTED+++++              802
+++++SS"ORM1"I"T                                                       803
SS"FCRM1"I"T                                                           804
PF"FCRM1                                                               805
&1&2 &1S BE&2&2ER &2C WEAR CU&2                                        806
PS"                                                                    807
+++++THIS PS FUNCTICN HAS NULL STRING+++++                             808
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                          809
RS                                                                     810
DS"FCRM1"FCRM1 IS REDEFINED CCES NCT CCNTAIN MARKERS                   811
PS"                                                                    812
+++++THIS PS FUNCTICN HAS NULL STRING+++++                             813
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                          814
RS                                                                     815
SS"FCRM1"NCN-EXISTING-1"NCN-EXISTING-2"NCN-EXISTING-3                  816
+++++NC SEGMENTATICN CF THAT FCRM HAS CCCURED+++++                     817
+++++SS"FORM1"NCN-EXISTING-1"NCN-EXISTING-2"NCN-EXISTING-3              818
PS"                                                                    819
+++++THIS PS FUNCTICN HAS NULL STRING+++++                             820
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                          821
RS                                                                     822
PF FCRM2                                                               823
+++++ERRCR FUNCTICN ,CCMMA IS MISSING AFTER FIRST ARGUMENT+++++        824
+++++PF FCRM2                                                          825
PF"FCRM2                                                               826
THAN TC RUST CUT                                                       827
PS"                                                                    828
+++++THIS PS FUNCTICN HAS NULL STRING+++++                             829
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                          830
RS                                                                     831
PF"UNDEFINED FCRM                                                      832
+++++ THERE IS NCT SUCH A FCRM IN STCRE+++++                           833
+++++PF"UNDEFINED FCRM                                                 834
PS"                                                                    835
+++++THIS PS FUNCTICN HAS NULL STRING+++++                             836
```

```
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                  837
RS                                                                             838
CL"FCRM1                                                                       839
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                              840
+++++FCRM WITH NC MARKERS ,ARGUMENTS CF CL FUNCTICN IGNCRED+++++               841
PS FCRM1 IS REDEFINED DCES NCT CCNTAIN MARKERS .                               842
+++++ERRCR FUNCTICN*****PS FCRM1 IS REDEFINED DCES NCT CONTAIN MARKERS          843
CL"FCRM1                                                                       844
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                              845
+++++FCRM WITH NC MARKERS ,ARGUMENTS CF CL FUNCTICN IGNCRED+++++               846
PS"FCRM1 IS REDEFINED DCES NCT CCNTAIN MARKERS                                 847
FCRM1 IS REDEFINED DCES NCT CCNTAIN MARKERS                                    848
PS"                                                                            849
+++++THIS PS FUNCTICN HAS NULL STRING+++++                                     850
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                  851
RS                                                                             852
DA                                                                             853
PS"                                                                            854
+++++THIS PS FUNCTICN HAS NULL STRING+++++                                     855
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                  856
RS                                                                             857
LN"STRING                                                                      858
+++++THERE ARE NC FCRMS IN STCRE+++++                                          859
PS"                                                                            860
+++++THIS PS FUNCTICN HAS NULL STRING+++++                                     861
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                  862
RS                                                                             863
DA                                                                             864
PS"                                                                            865
+++++THIS PS FUNCTICN HAS NULL STRING+++++                                     866
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                  867
RS                                                                             868
TN                                                                             869
PS"                                                                            870
+++++THIS PS FUNCTICN HAS NULL STRING+++++                                     871
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                  872
RS                                                                             873
EQ A2"A3"A4"A5                                                                 874
```

```
+++++ERROR FUNCTION NOTHING HAPPENED *****TRY AGAIN+++++
+++++EQ A2"A3"A4"A5
PS"
+++++THIS PS FUNCTION HAS NULL STRING+++++
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++
RS
EQ"A2"A3"A4
+++++ERROR FUNCTION NOTHING HAPPENED *****TRY AGAIN+++++
+++++EQ"A2"A3"A4
PS"
+++++THIS PS FUNCTION HAS NULL STRING+++++
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++
RS
EQ"A2"A3
+++++ERROR FUNCTION NOTHING HAPPENED *****TRY AGAIN+++++
+++++EQ"A2"A3
PS"
+++++THIS PS FUNCTION HAS NULL STRING+++++
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++
RS
EQ"A2
+++++ERROR FUNCTION NOTHING HAPPENED *****TRY AGAIN+++++
+++++ERROR FUNCTION NOTHING HAPPENED *****TRY AGAIN+++++
+++++EQ"A2
PS"
+++++THIS PS FUNCTION HAS NULL STRING+++++
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++
RS
EQ
+++++ERROR FUNCTION NOTHING HAPPENED *****TRY AGAIN+++++
+++++EQ
PS"
+++++THIS PS FUNCTION HAS NULL STRING+++++
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++
RS
EQ"A"A""ERROR
PS"
+++++THIS PS FUNCTION HAS NULL STRING+++++
```

```
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                          913
RS                                                                     914
EQ A2"A3"A4"A5                                                         915
+++++ERRCR FUNCTICN NCTHING HAPPENED *****TRY AGAIN+++++               916
+++++EQ A2"A3"A4"A5                                                    917
PS"                                                                    918
+++++THIS PS FUNCTICN HAS NULL STRING+++++                             919
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                          920
RS                                                                     921
EQ"A2"A3"A4                                                            922
+++++ERRCR FUNCTICN NCTHING HAPPENED *****TRY AGAIN+++++               923
+++++EQ"A2"A3"A4                                                       924
PS"                                                                    925
+++++THIS PS FUNCTICN HAS NULL STRING+++++                             926
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                          927
RS                                                                     928
EQ"A2"A2"VALUE FCURTH"ERRCR                                            929
PS"VALLE FCURTH                                                        930
VALUE FCURTH                                                           931
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                          932
RS                                                                     933
EQ"""VALUE FCURTH"ERRCR                                                934
PS"VALLE FCURTH                                                        935
VALUE FCURTH                                                           936
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                          937
RS                                                                     938
DS"FCRM1"NCT NULL                                                      939
DS"FCRM2                                                               940
+++++THIS STRING HAS NULL VALUE+++++                                   941
+++++DS"FCRM2                                                          942
DS"FCRM3"                                                              943
+++++THIS STRING HAS NULL VALUE+++++                                   944
+++++DS"FCRM3"                                                         945
EQ"""VALUE FCURTH"                                                     946
PS"VALLE FCURTH                                                        947
VALUE FCURTH                                                           948
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                          949
RS                                                                     950
```

```
CL"FORM3                                                                951
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                       952
+++++CALL TO A FORM WITH NULL VALUE+++++                                953
+++++CL"FORM3                                                           954
EQ"A2"A2""ERROR                                                         955
PS"                                                                     956
+++++THIS PS FUNCTION HAS NULL STRING+++++                              957
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                           958
RS                                                                      959
EQ"A2"A3"ERROR"VALUE FIFTH                                              960
PS"VALUE FIFTH                                                          961
VALUE FIFTH                                                             962
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                           963
RS                                                                      964
EQ""A3"ERROR"VALUE FIFTH                                                965
PS"VALUE FIFTH                                                          966
VALUE FIFTH                                                             967
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                           968
RS                                                                      969
CL"FORM2                                                                970
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                       971
+++++CALL TO A FORM WITH NULL VALUE+++++                                972
+++++CL"FORM2                                                           973
CL"FORM1                                                                974
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                       975
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++        976
EQ""NOT NULL"ERROR"VALUE FIFTH                                          977
PS"VALUE FIFTH                                                          978
VALUE FIFTH                                                             979
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                           980
RS                                                                      981
CL"FORM1                                                                982
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                       983
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++        984
CL"FORM3                                                                985
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                       986
+++++CALL TO A FORM WITH NULL VALUE+++++                                987
+++++CL"FORM3                                                           988
```

```
CL"FORM2                                                                    989
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                           990
+++++CALL TO A FORM WITH NULL VALUE+++++                                    991
+++++CL"FORM2                                                               992
EQ"NOT NULL""ERROR"                                                         993
PS"                                                                         994
+++++THIS PS FUNCTION HAS NULL STRING+++++                                  995
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                               996
RS                                                                          997
+++++TRAC JOB FINISHED THANK YOU GOOD-BYE,YEIA-XARA COSTAS+++++             998
                  PROCESSOR  FULL FROM AN ACTIVE  RS FUNCTION               999
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                              1000
PS"                                                                        1001
+++++THIS PS FUNCTION HAS NULL STRING+++++                                 1002
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                              1003
RS                                                                         1004
RS                                                                         1005
+++++VALUE OF ACTIVE RS EXCEEDS LIMITS PROCESSOR FULL+++++                 1006
+++++CARD1  RS                                                  +++++      1007
+++++                    CARD  RS                               +++++      1008
+++++                                          CARD3   RS      +++++      1009
+++++                                                          +++++      1010
+++++CARD 4    RS                                              +++++      1011
+++++                    CARD 5   RS                           +++++      1012
+++++                                       CARD   6   RS      +++++      1013
+++++                                                          +++++      1014
+++++         CARD   7   RS                                    +++++      1015
+++++                         CARD  8   RS                     +++++      1016
+++++                                          CARD  9  RS+++++           1017
+++++                                                            +++++      1018
+++++        CARD  10   RS                                       +++++      1019
+++++                           CARD  11                         +++++      1020
+++++                                        CARD   12          +++++      1021
+++++                                                           +++++      1022
+++++))"   SR "SR MORF LLUF ROSSECORP"SD "SP                              1023
                  PROCESSOR  FULL FROM A  NEUTRAL RS  FUNCTION             1024
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                              1025
+++++NEU VALUE OF META=   2          PI=   8+++++                          1026
```

```
+++++PROCESSOR FULL,EXECUTION STOPS+++++                                    1027
+++++PI=   2+++++                                                           1028
+++++)"                        31DRAC                          +++++        1029
+++++                                          21  DRAC        +++++        1030
+++++                                                          +++++        1031
+++++ 11   DRAC                                                +++++        1032
+++++                         Ó1   DRAC                        +++++        1033
+++++                                              9 DRAC      +++++        1034
+++++                                                          +++++        1035
+++++    8 DRAC                                                +++++        1036
+++++                              7 DRAC                      +++++        1037
+++++                                              6 DRAC      +++++        1038
+++++                                                    K     +++++        1039
+++++ KDRAC                                                    +++++        1040
+++++                   4DRAC                                  +++++        1041
+++++                                      3 DRAC              +++++        1042
+++++                                                          +++++        1043
+++++      2DRAC                                               +++++        1044
+++++                            1DRAC     "SP                              1045
                  PROCESSOR  FULL FROM AN ACTIVE  CL FUNCTION                1046
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++                                1047
+++++THIS PS FUNCTION HAS NULL STRING+++++                                   1048
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++                                1049
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                            1050
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++             1051
+++++VALUE OF ACTIVE CL EXCEEDS LIMITS PROCESSOR FULL+++++                   1052
+++++£(PS,ää(RS)£(CL,TEST FOR CL)))"  CARD1-CL                   +++++      1053
+++++                                        LC ROF TSET"LC      +++++      1054
+++++                                                  11  DRA+++++        1055
+++++C                                                          +++++        1056
+++++        01DRAC                                            +++++        1057
+++++                                  9DRAC                   +++++        1058
+++++                                                8DRAC     +++++        1059
+++++                                                          +++++        1060
+++++           7DRAC                                          +++++        1061
+++++                                      6DRAC               +++++        1062
+++++                                                    5D+++++            1063
+++++RAC                                                       +++++        1064
```

```
+++++                4DRAC                                                +++++          1065
+++++                                    3DRAC                            +++++          1066
+++++                                                          2   DR+++++          1067
+++++AC                                                                     +++++          1068
+++++                        1DRAC     "SP "SP                                          1069
                   PRCCESSCR  FULL FRCM A  NEUTRAL CL  FUNCTICN                          1070
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++                                           1071
+++++THIS PS FUNCTICN HAS NULL STRING+++++                                              1072
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                           1073
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                                       1074
+++++FCRM WITH NC MARKERS ,ARGUMENTS CF CL FUNCTICN IGNCRED+++++                        1075
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                                       1076
+++++FCRM WITH NC MARKERS ,ARGUMENTS CF CL FUNCTICN IGNORED+++++                        1077
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                                       1078
+++++FCRM WITH NC MARKERS ,ARGUMENTS CF CL FUNCTICN IGNCRED+++++                        1079
+++++NEL VALUE CF META=  16          PI=  58+++++                                       1080
+++++PRCCESSCR FULL,EXECUTICN STCPS+++++                                               1081
+++++PI=  16+++++                                                                      1082
+++++&&(CL,TEST-CL))"                                                      3+++++      1083
+++++DRAC                                                                  +++++      1084
+++++                2DRAC                                                 +++++      1085
+++++                                    1DRAC                             +++++      1086
+++++                                                  5DRAC               +++++      1087
+++++                                                                      +++++      1088
+++++  4DRAC                                                               +++++      1089
+++++                           3DRAC                                      +++++      1090
+++++                                              2DRAC                   +++++      1091
+++++                                                                    1D+++++      1092
+++++RAC                                                                   +++++      1093
+++++          5DRAC                                                       +++++      1094
+++++                              4DRAC                                   +++++      1095
+++++                                                  3DRAC               +++++      1096
+++++                                                                      +++++      1097
+++++     2DRAC                                                            +++++      1098
+++++                           1DRAC        "SP                                      1099
                   GARBAGE   CCLLECTICN                                                1100
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                          1101
PS"                                                                                   1102
```

```
+++++THIS PS FUNCTION HAS NULL STRING+++++                                1103
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                             1104
RS                                                                       1105
+++++NEW VALUE OF META= 926          PI= 961+++++                        1106
PS"TEST FOR ORDINARY CHARACTER WHEN TRAC PROCESSOR FULL                  1107
TEST FOR ORDINARY CHARACTER WHEN TRAC PROCESSOR FULL                     1108
+++++NEW VALUE OF META= 834          PI= 926+++++                        1109
+++++NEW VALUE OF META= 742          PI= 834+++++                        1110
+++++NEW VALUE OF META= 650          PI= 742+++++                        1111
+++++NEW VALUE OF META= 558          PI= 650+++++                        1112
+++++NEW VALUE OF META= 466          PI= 558+++++                        1113
+++++NEW VALUE OF META= 374          PI= 466+++++                        1114
+++++NEW VALUE OF META= 282          PI= 374+++++                        1115
+++++NEW VALUE OF META= 190          PI= 282+++++                        1116
+++++NEW VALUE OF META=  98          PI= 190+++++                        1117
+++++NEW VALUE OF META=   6          PI=  98+++++                        1118
PS"                                 CARD1                               1119
                                     CARD2                               1120
                                      CARD3                              1121
                                       CARD4                             1122
                                        CARD5                            1123
                                         CARD6                           1124
                                          CARD7                          1125
                                           CARD8                         1126
                                            CARD9                        1127
                                             CARD10                      1128
                                              CARD13                     1129
                                                                         1130
                                CARD1                                    1131
                                 CARD2                                   1132
                                  CARD3                                  1133
                                   CARD4                                 1134
                                    CARD5                                1135
                                     CARD6                               1136
                                      CARD7                              1137
                                       CARD8                             1138
                                        CARD9                            1139
                                         CARD10                          1140
```

```
                                        CARD13                                  1141
                                                                                1142
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                   1143
RS                                                                              1144
+++++TRAC JCB FINISHED THANK YCU GCCD-BYE,YEIA-XARA CCSTAS+++++                  1145
```

PART IIC:

T R A C L A N G U A G E :
T R A C A P P L I C A T I O N S

PART II: TRAC LANGUAGE

C. TRAC APPLICATIONS

IIC.1 Introduction

In this part some TRAC programs are included. They are not intended to be sophisticated since their main purpose is to test the TRAC interpreter which contains the TRAC scanning algorithm and the implemented TRAC primitive functions. There are some programs that could actually be written in a simpler way and yet provide the same results. The elaborate way in programming is chosen sometimes to prove the efficiency of the implemented system even in some awkward situations.

For more efficient programming in the TRAC system one should include in the implementation only the main primitive TRAC functions and then add some specialized primitive functions or user procedures of local interest. In this way the system would operate more efficiently for a particular installation.

The programs presented here are divided into groups, each referring either to one function or to a set of functions. They were written during the development of the present implementation in the following way:- The first group refers to the main TRAC primitive functions and it was written after these functions had been added to the TRAC system, which at the beginning contained only the TRAC algorithm. The TRAC algorithm together with the main primitive functions could then be used as an interpreter to interpret and execute procedures written in the TRAC language and including only these main primitive functions.

All other implemented functions were added to the system one at a time in the form of subroutines. This made it possible to add new

functions without affecting the rest of the system. Each new function had to be tested. A new group of programs was then included to test it. This set of programs also included the added function and the functions which had been implemented already. The order in the group of programs shows the order in which the TRAC functions were implemented.

Some programs from each group are described fully. The remaining ones could be checked from a terminal in interactive mode. The trace on function would be helpful for following the flow of execution on the neutral string and for understanding the logic of each program.

The listing for all programs is given first and it is followed by their execution in the same order. A title for each program facilitates reference between listing, execution and explanations given for each of them. All programs will be referred to by their title subsequently.

IIC.2 Execution of TRAC Programs

It has been mentioned already that at the beginning and at the completion of every processing cycle the TRAC "idling procedure", £(PS,£(RS)) is automatically loaded into the TRAC processor, and that therefore any processing is done in the second argument of the print string function. In fact all TRAC computations are made on functions nested within the argument string of some other function.

The idling procedure is loaded on the active string and is scanned from left to right and the evaluated characters are moved to the neutral string. The read string function is replaced by its value which is any character string read from the tele-typewriter. This value is scanned again according to the TRAC algorithm and all evaluated characters are moved to the neutral string where they form the evaluated functions and

then the appropriate action for each function is taken. When there is no further input string to be evaluated and all functions in the neutral string have been executed, the print string function of the idling procedure prints out any character remaining on the neutral string and the TRAC processor is reinitialized by reloading the idling procedure.

Every time the idling procedure is loaded, the TRAC processor asks for data when the read string function is evaluated. The data, given from the teletypewriter, are evaluated by the TRAC algorithm. When no more data are available, the moment the read string function of the idling procedure is executed, the null string is printed out, since this is the one which replaces the rs function, and the idling procedure is reloaded to ask subsequently for data.

In the present implementation, it is possible for the user to prevent the loading of the idling procedure. This is necessary when his particular job has finished and he has no more data for processing. Then, as the appropriate message says, he can type at the first position of his input string, a special character (here the apostrophe is chosen) and execution terminates. For a subsequent job the TRAC interpreter is called and execution begins always with the evaluation of the idling procedure.

The executed TRAC programs follow. The explanations given for each of the examples which are examined in detail are in some cases preceded by the description of the flow of execution and the steps taken in the neutral string in the order in which they occur.

IIC.3 Examples of TRAC Programs Executed

IIC.3.1 Example Test-1 functions RS, PS, CL, DS, SS

(a) Flow of execution from the neutral stack

aPScaRSe$_1$

aPScnPSc - - - ... - - - e$_2$

aPScnPSc***...***TEST-1 FUNCTIONS RS*PS*CL*DS*SS***...***e$_3$

aPScaDScAlcaRSe$_4$

aPScaDScAlcOVER THE M1 AND OVER THE WAVES(,)UNDER THE M2 AND UNDER THE M3e$_5$

aPScaSScAlcM1cM2cM3e$_6$

aPScaDScA2c£(CL,A1,MOUNTAINS,FOUNTAINS,GRAVES)e$_7$

aPScaPSc£(CL,A2)e$_8$

aPScaPScnCLcA2e$_9$

aPScaPSc£(CL,A1,MOUNTAINS,FOUNTAINS,GRAVES)e$_{10}$

aPScaPScaCLcA2e$_{11}$

aPScaPScaCLcA1cMOUNTAINScFOUNTAINScGRAVESe$_{12}$

aPScaPScOVER THE MOUNTAINS AND OVER THE WAVES,UNDER THE FOUNTAINS AND UNDER THE GRAVESe$_{13}$

aPScaPSc - - - ... - - - e$_{14}$

aPSce$_{15}$

(b) Description of execution

In this and all the following examples the idling procedure is loaded in the active string, and execution begins with the print string function. The evaluated functions are described in the order in which they are executed from the neutral string.

(1) When the read string function of the idling procedure is executed it causes the processor to accept input from the typewriter. This is the value of an active function. It replaces the function in the

active string, the scanning pointer is set to its first character and the scanning process is repeated from left to right. The input here consists of five cards up to the apostrophe, the terminating character.

(2) The first evaluated function of the input is the print string function ££(PS,----------). This prints out one line of "-" used to separate each example from the following one. After its evaluation, the ps function is deleted from the neutral string since it is a null-valued function.

(3) The following function is £(PS,**** ... TEST-1 FUNCTIONS RS*PS*CL *DS*SS***...*). It prints out ** ... TEST-1 FUNCTIONS RS*PS*CR*DS*SS ****...*, which is the title for the particular example and it is deleted from the neutral string as a null-valued function.

(4) The expression £(DS,A1,£(RS)) follows. This is an example of the "define string" primitive function. Here the third argument, which is the form recorded, is another function, the "read string" function, which appears in the active form. It is replaced by its value, which is the character string.

OVER THE M1 AND OVER THE WAVES ((,)) UNDER THE M2 AND UNDER THE M3

This string is evaluated by the scanning algorithm since it is the value of an active function. It contains no other functions and it remains unchanged, only one of the two pairs of parentheses which enclose the comma being removed during the scanning process.

(5) The define string function now causes recording of the evaluated string OVER THE M1 AND OVER THE WAVES (,) UNDER THE M2 AND UNDER THE M3 in the memory under the name A1. The ds function has null value and is therefore deleted, after being executed, from the neutral string and the scanning pointer is set to the first character of the next function in the active string.

(6) The next function £(SS,A1,M1,M2,M3) is an example of the segment string function. The form named A1 is taken from storage and is scanned

from left to right for occurrences of the string M1. A substring is found matching M1. The location of the match is marked and the matching substring is excluded from further action, creating a segment gap given the ordinal value one. There is only one match for the string M1. The same process is repeated for the arguments M2 and M3. At the end, three segment gaps are created, one for each argument, M1, M2 and M3, and they are marked with markers of ordinal value 1, 2 and 3 respectively. At the end the marked form A1, along with its pointers and ordinal identifiers for the segment gaps, is put back into storage with the name A1. After the segment string function is executed, the form A1 is stored as the string OVER THE m_1 AND OVER THE WAVES (,) UNDER THE m_2 AND UNDER THE m_3 where the m_i represent the internal markers and the index i denotes their ordinal value. The ss function has null value and after its execution is deleted from the neutral string and the scanning pointer set to the first character of the next function in the active string.

(7) The expression £(DS,A2,(£(CL,A1,MOUNTAINS,FOUNTAINS,GRAVES))) is another example where the define string function is used to store not text but another function. This function £(CL,A1,MOUNTAINS,FOUNTAINS, GRAVES) is protected from evaluation by a pair of parentheses, which is known as the quote mode in TRAC. Thus another form named A2 is created in storage which has as value the previous string which happens here to be another function. This ds function is replaced by its null value in the neutral string and the scanning process continues from the next character in the active string.

(8) The next expression £(PS,(£(CL,A2))) demonstrates the use of the quote mode in TRAC language. During the scanning process the enclosing pair of parentheses protects the call function £(CL,A2) from evaluation. The string £(CL,A2) is the second argument of the print string function and it is printed out without any alteration. Thus the result of the above expression is the printed string £(Cl,A2) as it is

shown in the execution sheet. The null value ps function is deleted from the neutral string when executed and the scanning continues from the next character in the active string.

(9) The following expression £(PS,££(CL,A2)) is an example of the neutral mode. This is a nested expression and the inner function ££(CL,A2) is evaluated first. This is a function in neutral mode which has a value. Since the function is in neutral mode its value is not evaluated again. The scanning in the active string continues from the next character but the neutral function is replaced by its value in the neutral string.

(10) The second argument of the previous function is replaced by its value. This value is the stored form A2. The form A2 remains unaltered since there are no arguments in the previous call function. The final result of the expression .9 is the print out of the string £(CL,A1,MOUNTAINS,FOUNTAINS,GRAVES) as it is shown in the execution sheet. The null-valued ps function is deleted from the neutral string and the scanning pointer is set to the next character in the active string.

(11) The following expression £(PS,£(CL,A2)) is an example of the active form. It is another nested expression and the inner function £(CL,A2) is evaluated first. The result of the active call function is to place the string value of the call, namely the form A2 which is £(CL,A1,MOUNTAINS,FOUNTAINS,GRAVES), in the former location of £(CL,A2) with an expansion (or a closing up in some other cases) of the surrounding strings in the active string. Then the active function £(CL,A2) is deleted from the neutral string and the value of this function is evaluated in the neutral string.

(12) The value of the previous function is another active cl function, £(CL,A1,MOUNTAINS,FOUNTAINS,GRAVES). When it is evaluated it is replaced by its value in the active string. This value is generated by

bringing the form A1 from storage and filling the segment gaps of ordinal value 1 with string MOUNTAINS, the gaps of ordinal value 2 with string FOUNTAINS, and the gaps of ordinal value 3 with string GRAVES. When the value which is the string OVER THE MOUNTAINS AND OVER THE WAVES(,) UNDER THE FOUNTAINS AND UNDER THE GRAVES is evaluated again the comma is unquoted.

(13) The evaluated value string is the second argument of a print string function. It is printed out and the result is shown in the execution sheet. It is the string OVER THE MOUNTAINS AND OVER THE WAVES, UNDER THE FOUNTAINS AND UNDER THE GRAVES, then the null-valued print string function is deleted from the neutral string and the scanning pointer is set to the next character in the active string.

(14) The last function is another print string function and prints out one line with "-" and then is deleted from the neutral string.

(15) The final print string function belongs to the idling procedure. It prints nothing out since the second argument is null. It is deleted from the neutral string and the idling procedure is loaded in the active string for subsequent processing.

IIC.3.2 Example Test-3 functions RS, PS, CL, DS, SS

(a) Flow of execution from the neutral string

aPScaRSe$_1$

aPScnPSc - - - ... - - - e$_2$

aPScnPSc***...***TEST-3 FUNCTIONS RS*PS*CL*DS*SS***...***e$_3$

aPScaDScPOEMc£(RS)e$_4$

aPSce$_5$

aPScaRSe$_6$

aPSc£(CL,POEM)e$_7$

aPSce$_8$

aPScaRSe$_{9}$

aPScnCLcPOEMe$_{10}$

aPSc£(RS)e$_{11}$

aPSce$_{12}$

aPScaRSe$_{13}$

aPScaCLcPOEMe$_{14}$

aPScaRSe$_{15}$

aPScWHEN I THE DAWN USED TO ADMIRE(,)AND PRAISED THE COMING DAYe$_{16}$

aPSce$_{17}$

aPScaRSe$_{18}$

aPScaDScPOEMcaCLcPOEMe$_{19}$ V

aPScaDScPOEMcaRSe$_{20}$

aPScaDScPOEMcWHEN I THE DAWN USED TO ADMIRE(,)AND PRAISED THE COMING DAYe$_{21}$

aPSce$_{22}$

aPScaRSe$_{23}$

aPScaSScPOEMcDAWNcPRAISEDcADMIREe$_{24}$

aPSce$_{25}$

aPScaRSe$_{26}$

aPScnCLcPOEMe$_{27}$

aPScWHEN I THE USED TO (,)AND THE COMING DAYe$_{28}$

aPScaRSe$_{29}$

aPScaCLcPOEMc1c2c3e$_{30}$

aPScWHEN I THE 1 USED TO 3,AND 2 THE COMING DAYe$_{31}$

aPScaRSe$_{32}$

aPScaCLcPOEMcMARKER1cMARKER2cMARKER3e$_{33}$

aPScWHEN I THE MARKER1 USED TO MARKER3,AND MARKER2 THE COMING DAYe$_{34}$

aPScaRSe$_{35}$

aPScaCLcPOEMcDAWNcPRAISEDcADMIREe$_{36}$

aPScWHEN I THE DAWN USED TO ADMIRE,AND PRAISED THE COMING DAYe$_{37}$

aPScaRSe$_{38}$

(b) Description of execution

The interactive mode of execution is demonstrated in this example. The idling procedure is loaded several times. In steps 1, 6, 9, 13, 15, 18, 23, 26, 29, 32, 35, 38 the TRAC processor asks for input to continue. When no further processing is desirable, the terminating character is typed in the first column, in accordance with the message from the TRAC processor. If data are typed, they replace the read string function of the idling procedure and they are evaluated by the TRAC algorithm.

The description of the most significant steps follows.

(4) The expression £(DS,POEM,(£(RS))) creates a form in storage. This form has the name POEM and a value string the function £(RS) which is not evaluated since it is protected by a pair of parentheses. This allows a function to be stored as a form and given a name.

(7) The expression (£(CL,POEM)) is in quote mode. It replaces the read string function of the idling procedure and the result printed is £(CL,POEM) since the protecting pair of parentheses is removed during evaluation.

(10) The typed data is a neutral function ££(CL,POEM). It is replaced by its value in the neutral string.

(11) The value of the above is the form named POEM which is the string £(RS). Here this string is the second argument of the print string function and the result printed is £(RS).

(14) The input is an active function £(CL,POEM). It is replaced by its value, which is the function £(RS), in the active string and then it is deleted from the neutral string.

(15) The above value, £(RS), is another active function. It is replaced by its value, which is the string read from the typewriter WHEN I THE DAWN USED TO ADMIRE((,)) AND PRAISED THE COMING DAY, in the active string and is deleted from the neutral string.

(16) The above value is evaluated. It remains unchanged since it does not contain any function, only one pair of parentheses enclosing the comma is removed. The print string function causes print-out of the text WHEN I THE DAWN USED TO ADMIRE(,) AND PRAISED THE COMING DAY.

(19) The example £(DS,POEM,£(CL,POEM)) is a redefinition of the form POEM which was defined and stored previously as the character string £(RS). This expression is the string typed from the typewriter when the TRAC processor asks for more data. It shows the flexibility of the TRAC language in dealing with text or procedures at any time. This is a nested procedure which replaces the rs function of the idling procedure. The inner function £(CL,POEM) is evaluated first. It is an active function. It is replaced by its value, £(RS), in the active string and is deleted from the neutral string.

(20) The value function, £(RS), is an active one. It is replaced by its value, the character string typed from the typewriter WHEN I THE DAWN USED TO ADMIRE ((,)) and PRAISED THE COMING DAY, in the active string and it is deleted from the neutral string.

(21) The above value is evaluated. It remains unchanged, since it does not contain any function, only one pair of parentheses enclosing the comma is removed. The evaluated string is the third argument of the define string function. The old form POEM is erased from memory and a new one is created with the above string as content. Then the ds function is deleted from the neutral string.

(24) The £(SS,POEM,DAWN,PRAISED,ADMIRE) creates a text macro. The form POEM is retrieved from memory and it is marked with markers of ordinal value 1, 2, 3 whenever a match with the substrings DAWN, PRAISED,

ADMIRE occurs. In this way the segment string function introduces the dummy or formal variables of the macro.

Four "macro-calls" follow in 27, 30, 33 and 36. The call function in 27 does not provide parameters to be substituted for the dummy variables. The internal markers in the form POEM are replaced by null strings. The result from the function ££(CL,POEM) is the string, WHEN I THE USED TO (,)AND THE COMING DAY. This is the value of a neutral function, it is not re-evaluated and the parentheses remain. In 30 the macro-call provides the parameters 1, 2 and 3 which replace the dummy variables DAWN, PRAISED and ADMIRE. In 33 the parameters provided are MARKER1, MARKER2 and MARKER3, and finally in 36 the parameters provided are DAWN, PRAISED and ADMIRE. Each of these functions is the second argument of the print string function of the idling procedure and their value after being evaluated is printed out in the following order:

(31) WHEN I THE 1 USED TO 3, AND 2 THE COMING DAY

(34) WHEN I THE MARKER1 USED TO MARKER3, AND MARKER2 THE COMING DAY

(37) WHEN I THE DAWN USED TO ADMIRE, AND PRAISED THE COMING DAY

This example shows the way of using facilities of the TRAC language interactively. Here the user changes the text macro during execution time by providing parameters to replace the dummy variables. Any text or procedural statement can be used as parameter and this is one of the features that make the language very flexible.

IIC.3.3 Example Test-10 functions RS, PS, CL, DS, SS

(a) Flow of execution from neutral stack

aPScRSe$_1$

aPScaPSc - - - ... - - - e$_2$

aPScnPSc***...***TEST-10 FUNCTIONS RS*PS*CL*DS*SS***...***e$_3$

aPScaDScQQQQcaRSe$_4$

aPScaDScQQQQc£(PS,HENCE(,)ALL YOU 2(,)AS SHORT AS ARE 1)e$_5$

aPScaCLcQQQQe$_6$

aPScaPScHENCE,ALL YOU 2,AS SHORT AS ARE 1e$_{6-1}$

aPScaPScQQQQ****nCLcQQQQe$_7$

aPScaPScQQQQ****£(PS,HENCE(,)ALL YOU 2(,) AS SHORT AS ARE 1)e$_8$

aPSce$_9$

aPScaRSe$_{10}$

aPScaSScQQQQcaRSe$_{11}$

aPScaSScQQQQc1c2e$_{12}$

aPScaDScQQQQcnCLcQQQQcaRSe$_{13}$

aPScaDScQQQQcnCLcQQQQcM1cM2e$_{14}$

aPScaDScQQQQc£(PS,HENCE(,)ALL YOU M2(,) AS SHORT AS ARE M1e$_{15}$

aPScaCLcQQQQe$_{16}$

aPScaPScHENCE ,ALL YOU M2,AS SHORT AS ARE M1e$_{17}$

aPScaPScQQQQ****nCLcQQQQe$_{18}$

aPScaPScQQQQ****£(PS,HENCE(,)ALL YOU M2(,)AS SHORT AS ARE M1e$_{19}$

aPSce$_{20}$

aPScaRSe$_{21}$

aPScaSScQQQQcaRSe$_{22}$

aPScaSScQQQQcM1cM2e$_{23}$

aPScaDScQQQQcnCLcQQQQcaRSe$_{24}$

aPScaDScQQQQcnCLcQQQQcMARKER1cMARKER2e$_{25}$

aPScaDScQQQQc£(PS,HENCE(,)ALL YOU MARKER2(,)AS SHORT AS ARE MARKER1)e$_{26}$

aPScaCLcQQQQe$_{27}$

aPScaPScHENCE,ALL YOU MARKER2,AS SHORT AS ARE MARKER1e$_{28}$

aPScaPScQQQQ****nCLcQQQQe$_{29}$

aPScaPScQQQQ****£(PS,HENCE(,)ALL YOU MARKER2(,)AS SHORT AS ARE MARKER1e$_{30}$

aPSce$_{31}$

aPScaRSe$_{32}$

aPScaSScQQQQcaRSe$_{33}$

aPScaSScQQQQcMARKER1cMARKER2e$_{34}$

aPScaDScQQQQcnCLcQQQQcaRSe$_{35}$

aPScaDScQQQQcnCLcQQQQcTHE NIGHTScVAIN DELIGHTSe$_{36}$

aPScaDScQQQQc£(PS,HENCE(,)ALL YOU VAIN DELIGHTS(,)AS SHORT AS ARE THE NIGHTS)e$_{37}$

aPScaCLcQQQQe$_{38}$

aPScaPScHENCE,ALL YOU VAIN DELIGHTS, AS SHORT AS ARE THE NIGHTSe$_{39}$

aPScaPScQQQQ****nCLcQQQQe$_{40}$

aPScaPScQQQQ****£(PS,HENCE(,)ALL YOU VAIN DELIGHTS(,)AS SHORT AS ARE THE NIGHTS)e$_{41}$

aPSce$_{42}$

(b) Description of execution

This is another example which is executed in the interactive mode. The first three steps are the usual ones for printing out the title of the example.

(4) The expression £(DS,QQQQ,£(RS)) is a define string function. It is a nested function and evaluation begins from the innermost function. A string is read from the typewriter and it replaces the active rs function in the active string.

(5) The input string is unquoted, since it is in the quote mode and a function is stored in memory under the name QQQQ. It is not evaluated since it is in the quote mode.

(6) The call function £(CL,QQQQ) is replaced by its value in the active string. This value is evaluated again. The stored form QQQQ is a print string function and when it is executed, the following string is printed HENCE,ALL YOU 2,AS SHORT AS ARE 1.

(7) In the expression £(PS,QQQQ****££(CL,QQQQ)) the call function is in the neutral mode. The form QQQQ is brought from storage and it

replaces the neutral function in the neutral string. This procedure allows the print-out of any stored form. It contains a print string function and a neutral call function.

(8) The form QQQQ is printed out for inspection. It is preceded by the string QQQQ**** mainly for identification purposes. Here the form QQQQ is printed out exactly as it is stored, that is, QQQQ****£(PS,HENCE (,)ALL YOU 2(,)AS SHORT AS ARE 1.

The following steps, 11-19, give as output the following two lines:

HENCE, ALL YOU M2,AS SHORT AS ARE M1

QQQQ****£(PS,HENCE(,)ALL YOU M2(,) AS SHORT AS ARE M1

and this is accomplished by executing the procedure £(SS,QQQQ,£(RS))£(DS, QQQQ,££(CL,QQQQ,£(RS)))£(CL,QQQQ)£(PS,QQQQ****££(CL,QQQQ)) (I).

In step 12 the segment string function creates a text macro where the dummy variables are read from the typewriter.

In step 14 the define string function redefines the form QQQQ. A call function provides the parameters which replace the dummy variables. The parameters are again read as a character string from the typewriter.

In step 16 the active call function causes the first line of the above output to be printed, and finally,

in step 19 the neutral call function causes the second line of the output to be printed out.

When the same procedure is typed again, when the above execution is completed and the TRAC processor asks for data to continue the processing, steps 22-30 and 33-41 repeat the processing in the same order as above. During each processing cycle, character strings typed when the processor asks for data are used as dummy variables or to replace the dummy variables parameters.

In more detail, in steps 20-30 the string M1, M2 provides the dummy variables; the string MARKER1, MARKER2 provides the parameters replacing

them and the two lines

HENCE, ALL YOU MARKER2, AS SHORT AS ARE MARKER1

QQQQ****£(PS,HENCE(,) ALL YOU MARKER2(,) AS SHORT AS ARE MARKER1)

are the output obtained.

Similarly in steps 33-41 the string MARKER1, MARKER2 provides the dummy variables, the string THE NIGHTS, VAIN DELIGHTS provides the parameters replacing them, and the two lines

HENCE, ALL YOU VAIN DELIGHTS, AS SHORT AS ARE THE NIGHTS

QQQQ****£(PS,HENCE(,) ALL YOU VAIN DELIGHTS(,) AS SHORT AS ARE THE NIGHTS)

are the output obtained.

In this example the procedure (I) is typed three times and causes the same processing to be repeated each time. The different data that are typed, when the processor asks for input, allow the user to obtain various results.

The following example demonstrates a more efficient method of handling the same problem in TRAC language.

IIC.3.4 Example Test-11 functions RS, PS, DS, SS, CL

In this example, only the steps which are different from the corresponding ones in example 10 are described.

(a) Flow of execution from neutral stack

aPScaDScQQQQc£(PS,HENCE(,)ALL YOU 2(,)AS SHORT AS ARE 1)£(PS,QQQQ****
££(CL,QQQQ))£(SS,QQQQ,£(RS))£(DS,QQQQ,££(CL,QQQQ,£(RS)))£(CL,QQQQ)e$_4$

aPScaCLcQQQQe$_5$

aPScaPScHENCE,ALL YOU 2,AS SHORT AS ARE 1e$_6$

aPScaPScQQQQ****nCLcQQQQe$_7$

aPScaPScQQQQ****£(PS,HENCE(,)ALL YOU 2(,)AS SHORT AS ARE 1)£(PS,QQQQ****
££(CL,QQQQ))£(SS,QQQQ,£(RS))£(DS,QQQQ,££(CL,QQQQ,£(RS))£(CL,QQQQ)e$_8$

aPScaSScQQQQcaRSe$_9$

aPScaSScQQQQclc2e$_{10}$

aPScaDScQQQQcnCLcQQQQcaRSe$_{11}$

aPScaDScQQQQcnCLcQQQQcM1cM2e$_{12}$

aPScaDScQQQQc£(PS,HENCE(,)ALL YOU M2(,)AS SHORT AS ARE M1)£(PS,QQQQ****
££(CL,QQQQ))£(SS,QQQQ,£(RS))£(DS,QQQQ,££(CL,QQQQ,£(RS)))£(CL,QQQQ)e$_{13}$

aPScaCLcQQQQe$_{14}$

(b) Description of execution

Define string expressions can be used to store procedures in the TRAC language. The procedure which was typed every time it was executed, in the previous example, is stored here as a form with the name QQQQ.

(4) The define string expression causes a procedure to be stored under the name QQQQ. This procedure is protected from evaluation since it is presented in the quote mode. The last function of the procedure is an active call function which during execution calls the procedure to be evaluated. This procedure can be seen as a do loop in which the first statement of its range is the print string function and the last the active call function referred to above.

(5) The procedure is put into action by the following active call function £(CL,QQQQ). This causes a copy of the procedure to be loaded into the active string. The procedure is executed with the evaluation of each of the functions it contains.

(6) The print string function causes the print-out of the string HENCE, ALL YOU 2, AS SHORT AS ARE 1

(7) The expression £(PS,QQQQ****££(CL,QQQQ)) causes the print-out of the procedure QQQQ as it is stored in memory. The string QQQQ**** precedes the form for identification purposes only. This result is shown in the execution sheet.

(8) The segment string function introduces the dummy variables 1,2 as they are read from the typewriter when the processor asks for input.

(10) The define string expression £(DS,QQQQ,££(CL,QQQQ,£(RS))) redefines the form QQQQ by replacing the dummy variables with the parameters M1, M2 as they are read from the typewriter when the processor asks again for input.

(14) The last action of the procedure is to call another copy of itself from the store into the active string for subsequent processing. Thus the £(CL,QQQQ) active call function transfers control to the beginning of the same procedure.

When the procedure QQQQ is executed subsequently the order of evaluation is the following: for the second time —

(a) HENCE, ALL YOU M2, AS SHORT AS ARE M1

(b) print-out of the form QQQQ

(c) M1, M2 as dummy variables

(d) MARKER1, MARKER2 as parameters

for the third time —

(a) HENCE, ALL YOU MARKER2, AS SHORT AS ARE MARKER1

(b) print-out of the form QQQQ

(c) MARKER1, MARKER2 as dummy variables

(d) THE NIGHTS, VAIN DELIGHTS as parameters

for the fourth time —

(a) HENCE, ALL YOU VAIN DELIGHTS, AS SHORT AS ARE THE NIGHTS

(b) print-out of the form QQQQ

This loop could be an endless one. The TRAC language provides facilities for exit from a DO loop. These are conditional functions like the equal function £(eq, X1, X2, T, F) or the greater than function £(gr, D1, D2, T, F).

In this example an unconditional loop occurs when the terminating character for the TRAC job is typed and read as data.

IIC.3.5 Example Test-14 functions RS, PS, CL, DS, SS

In this and the following examples, the initial steps of the idling procedure and of the title writing are omitted.

(a) Flow of execution from neutral stack

aPScaDScPOEMc£(DS,VERSEM,Y)e$_4$

aPScaPScPOEM*****nCLcPOEMe$_5$

aPScaPScPOEM*****£(DS,VERSEM,Y)e$_6$

aPScnSScPOEMcMcYe$_7$

aPScaPScPOEM*****nCLcPOEMe$_8$

aPScaPScPOEM*****£(DS,VERSE,)e$_9$

aPScaPScnCLcaRSe$_{10}$

aPScaPScnCLcPOEMclcTO ME ONE SAVAGE HUNTING SCENE(,)MY SOLE DELIGHT THE HEADLONG RACEe$_{11}$

aPScaPSc£(DS,VERSEl,TO ME ONE SAVAGE HUNTING SCENE(,)MY SOLE DELIGHT THE HEADLONG RACE)e$_{12}$

aPScaCLcaRSe$_{13}$

aPScaCLcPOEMclcTO ME ONE SAVAGE HUNTING SCENE(,)MY SOLE DELIGHT THE HEADLONG RACEe$_{14}$

aPScaDScVERSElcTO ME ONE SAVAGE HUNTING SCENE(,)MY SOLE DELIGHT THE HEADLONG RACEe$_{15}$

aPScaPScnCLcVERSEle$_{16}$

aPScaPScTO ME ONE SAVAGE HUNTING SCENE MY SOLE(,)DELIGHT THE HEADLONG RACEe$_{17}$

aPSce$_{18}$

(b) Description of execution

Here a define string expression is used to store a procedural statement. This procedural statement is converted to a procedural macro by the segment string operation. Then macro calls to the procedure can be made and this causes execution of the procedure with the

parameters inserted.

(4) The define string function £(DS,VERSEM,Y) is stored in memory under the name POEM. It is protected from evaluation since it is presented in the quote mode.

(6) The expression £(PS,POEM*****££(CL,POEM)) provides the means for checking the stored form. This form, preceded by the string POEM*****, is printed out for inspection. It can be seen in the execution sheet that this result is POEM*****£(DS,VERSEM,Y).

(7) The segment string function converts the function stored to a macro by providing the dummy variables M, Y which are replaced internally by markers of appropriate ordinal value.

(9) The expression £(PS,POEM*****££(CL,POEM)) is now a call to the macro POEM. It does not provide any parameters to replace the dummy variables. As can be seen in the execution sheet, the dummy variables are replaced by null strings and the result obtained is

POEM*****£(DS,VERSE,)

The following procedure,

£(PS,££(CL,£(RS)))£(CL,£(RS))£(PS,££(CL,VERSE1))

contains two macro calls to the procedural macro POEM. The parameters provided by the rs function are identical in both calls, but the first is a neutral call and is replaced by its value in the neutral string, which is printed out for inspection by a subsequent print string function, while the second is an active call and causes its value to be evaluated from the active string. This evaluation creates another form in store. Description in detail of the above procedure follows.

(11) The value of the rs function, which provides the parameters for the first call, is evaluated.

(12) The neutral call function is replaced by its value in the neutral string and this value is printed out since it is the second argument of a print string function. The result obtained is the

character string £(DS,VERSE1, TO ME ONE SAVAGE HUNTING SCENE(,) MY SOLE DELIGHT THE HEADLONG RACE). This is in fact the value of the first macro call which is not evaluated since its mode was neutral.

(14) The second macro call inserts the same parameters to replace the dummy variables but now the mode is active and causes execution of the procedural macro with the parameters inserted.

(15) The value of the macro call, which is identical with the above written value of the neutral call, is now evaluated in the active string. The result is a new form VERSE1 with content TO ME ONE SAVAGE HUNTING SCENE(,) MY SOLE DELIGHT THE HEADLONG RACE to be stored in memory. The expression £(PS,££(CL,VERSE1)) simply prints out the recorded form VERSE1. Its result is the string TO ME ONE SAVAGE HUNTING SCENE, MY SOLE DELIGHT THE HEADLONG RACE and this proves that the second macro call is executed in the way described previously.

At the end of this processing cycle the TRAC processor asks for input. Then a similar procedure, £(PS,££(CL,£(RS)))£(CL,£(RS))£(PS, ££(CL,VERSE2)), is typed in. The two macro calls are the same as previously only the inserted parameters change and this results in the recording of a new form in storage.

Steps 10-18 are repeated during this second processing and the corresponding results obtained are £(DS,VERSE2,AND FRANTIC HURRY OF THE CHASE TO START(,) PURSUE(,) AND BRING TO BAY) which is the result of the neutral macro call and AND FRANTIC HURRY OF THE CHASE TO START, PURSUE, AND BRING TO BAY, which is the result of the expression £(PS,££(CL,VERSE2)) after the execution of the active macro call £(CL,£(RS)).

IIC.3.6 Example Test-16 functions RS, PS, CL, DS, SS

(a) Flow of execution from neutral stack

aPScaDScEXcPROGRAM SHOWING THE USE OF A PROCEDUREe$_4$

aPSce$_5$

aPScaDScVERSE1cTHEN LET NOT WHAT I CANNOT HAVEe$_6$

aPSce$_7$

aPScaDScVERSE2cMY CHEER OF MIND DESTROYe$_8$

aPSce$_9$

aPScaDScVERSE3cWHILST THUS I SING(,)I AM A KING(,)ALTHOUGH A POOR BLIND BOYe$_{10}$

aPSce$_{11}$

aPScaDScPROCEDUREc££(PS,(GIVE NAME***...***))£(PS,£(CL,£(RS)))£(CL, PROCEDURE)e$_{12}$

aPSce$_{13}$

aPScaCLcPROCEDUREe$_{14}$

aPScnPScGIVE NAME***...***e$_{15}$

aPScaPScaCLcaRSe$_{16}$

aPScaPScaCLcEXe$_{17}$

aPScaPScPROGRAM SHOWING THE USE OF A PROCEDUREe$_{18}$

aPScaCLcPROCEDUREe$_{19}$

(b) Description of execution

This is another example of self-referencing and interaction during execution with TRAC. The expression ££(PS,(GIVE NAME***...***)) £(PS,£(CL,£(RS)))£(CL,PROCEDURE) (I) is recorded in storage under the name PROCEDURE. When this procedure is put into action with an active call function, £(CL,PROCEDURE), the form PROCEDURE is retrieved from store and is moved into the active string for evaluation. The last action of this procedure is to call another copy of itself from the store

into the active processing area. The process is a do loop which does not provide means for exit. It can be executed an unlimited number of times. In the example an unconditional exit occurs when the terminating character of TRAC processing is typed as data. In steps 4, 6, 8, 10 the forms EX, VERSE1, VERSE2, VERSE3 are defined and stored.

(12) The procedure, PROCEDURE, is now recorded. It has as content the character string (I), which is not evaluated, since it is presented in the quote mode. The above procedure allows the user to call forms previously recorded and inspect their content.

(14) The procedure is put into action by the active function £(CL,PROCEDURE). This is replaced by its value in the active string and is deleted from the neutral string.

(15) The first function of the procedure is the neutral print string function ££(PS,(GIVE NAME***...***)). Its result is the print-out of the string GIVE NAME***...***. After being executed the print string function is deleted from the neutral string.

(16) The next nested functional expression, £(PS,£(CL,£(RS))), is now evaluated. The innermost active read string function is evaluated first. The user is informed by the previous message GIVE NAME ***...*** and when the processor asks for data, he types the name of the form he wants to inspect.

(17) The name EX is typed and the corresponding form is called from memory.

(18) The form is evaluated and replaces the call function. The evaluated form is now the second argument of a print string function. The printed-out string PROGRAM SHOWING THE USE OF A PROCEDURE is in fact the form recorded in store under the name EX.

(19) The last function of the procedure is another call which causes a new copy of itself to be loaded and subsequently evaluated.

Steps 15-19 are repeated and the processing continues until the terminating character is typed as data.

The following lines show the interaction during execution of the above procedure. In the left-hand column are the data as typed by the user and in the right-hand column the corresponding computer responses are shown.

Data Typed by the User	Computer Responses
£(CL,PROCEDURE)	GIVE NAME**********
EX	PROGRAM SHOWING THE USE OF A PROCEDURE
VERSE1	GIVE NAME********** THEN LET WHAT I CANNOT HAVE
VERSE2	GIVE NAME********** MY CHEER OF MIND DESTROY
VERSE3	GIVE NAME********** WHILST THUS I SING, I AM A KING, ALTHOUGH A POOR BLIND BOY
£(PS,---...---)	GIVE NAME********** THERE IS NO SUCH FORM IN FSTORE TO BE CALLED
' (Quote)	GIVE NAME********** END OF TRAC JOB

In the cycle before the last, after the message GIVE NAME, a print string function is typed. Since it is a null-valued function it is deleted after its evaluation. The active call function asks for a form with null name which does not exist in store, and the appropriate response is given by the computer.

IIC.4 Group-2 Programs for Testing TN,TF Functions

Two examples from this group are described. The trace on function is a null-valued function which allows the trace of flow at execution time. After the evaluation of this function, whenever a function is formed in the neutral string it is printed out before it is

executed. The action caused by the trace on function stops when the trace off is evaluated. The trace off function is another null-valued function. Its only result is to stop the action caused by the trace on function.

IIC.4.1 Example Test-1 for functions TN,TF

The input value which replaces the read string function of the idling procedure contains two neutral print string functions and one active trace on function. During evaluation one line with the character "-" and another with the title of the example are printed out. The trace on function is evaluated and causes the TRAC processor to print out any function, which has been evaluated and placed in the neutral string, before it is executed.

The print string function of the idling procedure is now evaluated. Its second argument is null since its value consisted of null-valued functions. The trace on function causes the print-out of this function, which is PS", where " stands for the internal marker for comma which cannot be printed. This ps function is executed subsequently and prints out nothing, since its second argument is the null substring. Then the processor is loaded with the idling procedure.

The next function RS is printed before its execution. When executed it is replaced by the typed input character string. The trace on function prints the first evaluated function of this input, which is DS"1"£(PS,POOR SOUL(,) THE CENTRE OF MY SINFUL EARTH)£(SS,1,£(RS))£(DS,1,££(CL,1,£(RS))). This function is executed and records the form 1 as a procedure containing six functions.

The next function of the input, CL"1 as printed by the tn, loads a copy of the form 1 in the active string for evaluation. The order in which functions are evaluated is easily seen as they are printed,

before they are executed, by the trace on function. The printed functions are

PS" POOR SOUL, THE CENTRE OF MY SINFUL EARTH

RS

SS"1"SOUL"EARTH

RS

CL"1"EARTH"SOUL

DS"1"£(PS,POOR EARTH(,) THE CENTRE OF MY SINFUL SOUL)

£(SS,1,£(RS))£(DS,1,££(CL,1,£(RS)))

In fact these are the six functions of the procedure 1 in the order in which they are executed. The results obtained can be seen on the execution sheet. It is seen that by using segment and call operations the procedure is redefined at execution time.

The last function of the input is another active call function, CL"1, and it loads another copy of the redefined form 1 in the active string for evaluation.

The functions of the procedure 1 are evaluated again in the same order. The result obtained from the print string function POOR EARTH, THE CENTRE OF MY SINFUL SOUL is now different since the form 1 has been redefined.

When the RS function is evaluated and the processor asks for data the function £(TF) is typed in. It is printed out before it is executed, and after its evaluation no more functions are typed out from the neutral string since the trace off function cancels the execution of the trace on function. It is a null function and is deleted after being executed and thus the segment string function is not provided with arguments and the form 1 is not segmented.

When the next rs function is evaluated, another null-valued function is typed in. This is a print string function and it prints out another line with "-" and is deleted from the neutral string; thus the

call function is not provided with any arguments. The second processing cycle terminates and execution is finished by the typing of the terminating quote character.

IIC.4.2 Example Test-2 for Functions TN,TF

The first input string that replaces the read string function of the idling procedure is similar to the one in the previous example. Its last function is a trace on function.

When the idling procedure is loaded again another processing cycle begins and the trace on function causes the following functions to be printed out in the order in which they are evaluated:

```
RS
DS"BUILDLINE"1 AND HIS DOG
SS"BUILDLINE"1"
DS"THIS THING"ONE MAN
DS"WTM"1 WENT TO MOW,
SS"WTM"1"
TF
```

The RS function belongs to the idling procedure. It causes the input string to be read and evaluated. The define string function records the form BUILDLINE and stores its content 1 AND HIS DOG in memory. It is evaluated first, since it is the innermost function of the expression £(SS,BUILDLINE,1,£(DS,BUILDLINE,1,AND HIS DOG)). The segment string function which is evaluated next replaces the character 1 of the form BUILDLINE with a marker of ordinal value one. The next define string function creates another form with name, THIS THING, and content, ONE MAN. The following define string function records the form WTM and stores its content 1 WENT TO MOW,. It is the innermost function of the expression £(SS,WTM,1,£(DS,WTM,1 WENT TO MOW(,))). The

segment string function which follows replaces the character 1 of this form with an internal marker. The TF function cancels the trace operation and processing continues in the normal way.

Another form, WIMAM with content, WENT TO MOW A MEADOW., is defined. All defined functions are printed out after execution of a nested call and print function for each form. The call functions do not provide any arguments and if internal markers exist they are replaced by null.

The form BUILDLINE is printed out as the string AND HIS DOG since the marker which exists in place of 1 is replaced by null. The form THISTHING is printed out as ONE MAN. It remains unchanged, since it is not segmented. WENT TO MOW, is printed for the form WIM, 1 is again replaced by null and the form WIMAM is printed out without any change as WENT TO MOW A MEADOW. Finally execution is terminated by the typing of the quote in col. 1.

IIC.5 Group-3 Programs for Testing DA Function

The function DA erases all existing forms from storage. It can be used to allow the continuation of execution when the store is full and the forms defined are not needed any more. It is another null-valued function and its result is the same whether the mode is neutral or active. Two examples from this group are described in the following paragraphs.

IIC.5.1 Example Test-3 for Function DA

The rs function of the idling procedure causes a character string of five consecutive lines to be read from the typewriter and to be moved into the active string for processing.

The input is evaluated and two lines are printed out: the first is the one used to separate programs and consists of the character "-", and the second is the title for the program.

The following two define string forms cause the recording of two forms, namely VERSE1 and VERSE2. The content of these forms is a character string read from the typewriter since the third argument of both functions is a read string function. The only difference is that the rs function belonging to the first ds is in active form but the one belonging to the second ds is in neutral form. This makes the second ds more efficient since the value of the rs function passes immediately to the neutral string without further evaluation on the active string, as happens in the first case.

Subsequent segment string operations cause the segmentation of both forms VERSE1 and VERSE2 and the replacement of some of their substrings with segment gap indicators with appropriate ordinal values. Three segment gaps are created in VERSE1 and four in VERSE2.

Two nested print and call expressions follow. In the first of them, the call function provides the parameters LEST, WISE WORLD, and MOAN to replace the internal markers. The value of this function, after its evaluation, is the second argument of the print string function which causes the print-out of the string LEST THE WISE WORLD SHOULD LOOK INTO YOUR MOAN. This procedure has changed the input character string 1 THE 2 SHOULD LOOK INTO YOUR 3 to the above meaningful sentence.

Similarly the second nested print and call expression provides the second line of output, which is AND MOCK YOU WITH ME AFTER I AM GONE., and this is the changed form of the second read character string AND 1 YOU 2 ME 3 I AM 4.

The list name function lists the names of all existing forms and prefaces each name with the string *****. The output from this function

is *****VERSE1

*****VERSE2

THERE ARE NO FORMS IN STORE

since there are only two forms VERSE1 and VERSE2 stored.

The delete all function that follows erases all forms defined before its execution from storage. The three following expressions prove that the delete all function has, in fact, the expected result. The list name function gives the message, THERE ARE NO MORE FUNCTIONS, since all existing forms have been deleted. The two call functions contained in the following two nested expressions cannot find the forms VERSE1 and VERSE2 in storage, since they have been deleted by the da function and the appropriate message is given. Execution is terminated by the typing of the quote character in column 1.

IIC.5.2 Example Test-4 for Function DA

The rs function of the idling procedure is replaced by four consecutive lines. This is the input character string typed in from the typewriter. The first two lines are similar to the ones in the previous example.

The following define string function causes the recording of a character string read from the typewriter and evaluated from the active string, since the third argument is an active read string function. The name of the recorded string is A1 and its content the evaluated character string THROUGH MARKER2 OF MY LONG MARKER1 STAY, which is a meaningless string of characters. This string is segmented by a subsequent segment string operation. Two of its substrings are replaced by internal markers. According to their order in the segment string operation, the substring MARKER1 is replaced by an internal marker of ordinal value 1 and the substring MARKER2 by another one of ordinal value 2.

A procedure is defined. It is given the name PROGRAM and content the nested expression £(PS,£(CL,£(RS))). When this procedure is put into action by an active call function, £(CL,PROG), it loads a copy of its content into the active string. This value of the active call function is then evaluated. It is a nested functional expression and the internal read string function is evaluated first.

The read string function is replaced by its value, which is the string read from the typewriter A1,FRUITLESS,DISCONTENT. This string is evaluated since it is the value of an active function. The evaluated string now provides the parameters for the active call function. This function after the execution of the rs function has the form £(CL,A1,FRUITLESS,DISCONTENT).

The form A1 is brought from storage and the segment gap indicators are replaced by the arguments of the call function. The argument FRUITLESS replaces the marker with ordinal value one and the argument DISCONTENT replaces the marker with ordinal value two. This value replaces the active call function and is evaluated in the active string. The evaluated string is THROUGH DISCONTENT OF MY LONG FRUITLESS STAY and this is printed out since it is the second argument of a print string function.

The list names function which is executed next provides three lines of output *****A1

*****PROG

THERE ARE NO FORMS IN STORE

A1, PROG are the names of the stored forms, ***** is the string which precedes names and the message is printed since there are no other forms in storage.

The expression £(£(DA)CL,PROG) contains one delete all function and one call function. The delete function is executed first and erases the two forms A1 and PROG from storage. It is a null-valued

function and is replaced by its null value in the neutral string and the active call function is now evaluated.

Since the delete all function has been evaluated first, all forms are erased from storage. The TRAC processor cannot find the form called and it prints out the appropriate message meaning that the form PROG is not in storage any more.

Execution stops when the quote character in column 1 is read.

IIC.6 Group-4 Programs for Testing LN Function

The listing names, ln, is another null-valued function which provides a listing of the names of the forms which are stored in memory. It can be used mainly for checking purposes, to find out if all expected forms are stored or to provide information, since it is possible when the names of the forms stored are known to call them from store, print them out and see their structure and the processing action they can take.

IIC.6.1 Example Test-7 for function LN

The first time the rs function of the idling procedure is replaced by two lines of input typed from the typewriter. This input, after evaluation, causes the print-out of two lines, the separator line full of "-" characters and the title line. After this processing cycle the idling procedure is loaded again and when its rs function is executed, causes the TRAC processor to ask for input.

This input is the character string which is formed by the following three typed lines, and it is evaluated from the active string. The evaluated functions are described in the order in which they are executed.

The nested define string expression £(DS,A,ff(RS)) causes another

line of input to be read in and placed into the neutral string. This is the character string THESE DELIGHTS IF THOU CANST TAKE(,) MIRTH(,) WITH THEE I MEAN TO DIE, which now becomes the third argument of the define string function and is stored under the name A.

A subsequent segment string function replaces the string TAKE with an internal marker of ordinal value 1 and the string DIE with an internal marker of ordinal value 2 and replaces the old form A with the new segmented one.

The following define function stores a procedure in memory. The name of the procedure is B and its content one call function, £(CL,A, GIVE,LIVE). Three print string functions follow. The first has its second argument in the quote mode. This argument is printed out as £(CL,B) after the removal of the protecting pair of parentheses. The second one has its second argument in the neutral mode, ££(CL,B). The form B is brought from storage but it is not evaluated since this call is a neutral one. This print string function causes the line £(CL,A,GIVE,LIVE) to be printed out. The third one has its second argument in the active mode. This argument, £(CL,B), is replaced by its value, £(CL,A,GIVE,LIVE), in the active string. Now the form A is brought from storage, the internal markers are substituted by the corresponding arguments and the resulting value after another evaluation gives the argument of the print string function which is printed out and which is the following line THESE DELIGHTS IF THOU CANST GIVE, MIRTH, WITH THEE I MEAN TO LIVE, which is in fact the meaningless input character string changed to a meaningful result.

Four LN functions follow in active or neutral mode to prove that execution of a null-valued function is not affected by its mode. The first, £(LN,NAME), prints the result NAMEA,NAMEB,THERE ARE NO FORMS IN STORE. The string NAME, which is the second argument of the LN function,

precedes the names of the existing forms A and B. The message appears when the TRAC processor cannot find other forms in storage. The argument in the second LN function is NAME followed by three blanks and the result becomes NAME A, NAME B, which makes easier the reading of the listed names. The third is coded in the wrong way, the appropriate message is given and the function is printed out for inspection. Here a comma is missing after its first argument. In the fourth LN function the second argument is null and the listed names are not preceded by any string. The output contains only the listing of the form names and the terminating message when all names are printed out.

IIC.6.2 Example Test-4 for function LN

The first two lines of output are printed out as in the previous example. The idling procedure is loaded for the second time and the read string function is evaluated and an input string of three consecutive lines is read in. This string is now evaluated from the active string. The evaluated functions are described in the order of their execution.

The nested define string function €(DS,,€€(RS)) stores the string read in from the typewriter and gives to it a null name since its second argument is the null string. This null-named form has the content WHEN MARKER2 DO MARKER4 SING MARKER5 HEY MARKER1 A MARKER1 MARKER1(,) SWEET MARKER3RS MARKER3 THE MARKER5 SPRING MARKER4.

The segment string functions create five segment gaps and replace the substrings MARKER1, MARKER2, MARKER3, MARKER4 and MARKER5 by internal markers of ordinal value 1, 2, 3, 4 and 5 respectively.

A print and call nested function occurs. The call function is an active one which brings the null-named form from storage and replaces the internal markers of ordinal value 1, 2 and 3 by the arguments DING,

BIRDS and LOVE respectively. The remaining markers are replaced by null since there are no corresponding arguments.

The value of this active call function is evaluated and then printed out since it is the second argument of a print string function. The printed line of output is WHEN BIRDS DO SING HEY DING A DING DING, SWEET LOVERS LOVE THE SPRING.

Thus the above procedure has changed the meaningless input text to a meaningful output text.

A similar procedure is used subsequently to define a new form XY, to segment this form and create two segment gaps by replacing the substrings MARKER1 and MARKER2 with internal markers. These markers are replaced by the arguments of the call function which are THOUGH YOU DO and SO TRUE A FOOL IS and a meaningful output is obtained.

Four LN functions and their corresponding results are shown in the following table:

	Function	Results
(1)	£(LN,NAME)	NAME
		NAMEXZ
		THERE ARE NO FORMS IN STORE
(2)	£(LN,NAME)	NAME
		NAME XZ
		THERE ARE NO FORMS IN STORE
(3)	£(LN COMMA MISSING)	ERROR FUNCTION COMMA IS MISSING
		AFTER FIRST ARGUMENT
		LN COMMA MISSING
(4)	£(LN,)	XZ

In the first two, the names of the stored forms are printed out with the second argument, NAME, preceding them. In the second function, the second argument is the string NAME followed by two blanks, which help

readability. Since the first form has null name, nothing is printed in the place of the form name. The third function is coded in the wrong way. A message and the whole function are printed out for inspection. In the fourth function, the second argument is null and the names are printed out without any preceding string. In fact here only the name XZ is printed since the first form has null name.

IIC.7 Group-5 Programs for Testing PF Function

The print form, pf, function is used to print out the content of any form in storage. It provides means for identifying the internal markers by printing a special character followed by the ordinal value of the marker. This function makes it possible to inspect the content of a form even when this form has been segmented by a previous segment string operation.

IIC.7.1 Example Test-2 for function PF

The read string function of the idling procedure is replaced by the usual character input string which provides the output of the two first lines for the title.

The second time, when the idling procedure is loaded, its rs function is replaced by an input string made up of four consecutive lines read in from the typewriter. This is the value of the rs function and it is evaluated from the active string.

The define string function stores a character string of the three following read-in lines and names it TEXT. The form TEXT is segmented by the following segment string function and internal markers of ordinal value 1, 2, 3 and 4 replace the substrings of its content THAT, THE, IN and COINCIDE respectively.

The following print form function asks for the print-out of the form NAME and the TRAC processor responds with the message THERE IS NOT SUCH A FORM IN STORE, since no form named NAME was defined before the print form function was executed. A similar message is printed after the execution of the second print form function £(PF,TEXT). Although there is a form TEXT defined previously, the second argument of this function contains the string TEXT followed by three blanks and in fact no such form TEXTbbb has been defined. The following print form function £(PF,TEXT) is the correct one for printing out the content of the stored form TEXT. It can be seen, on the execution sheet, that the original input text has been altered. The marker with ordinal value 1 occurs two times which is the same as the number of occurrences of its corresponding argument substring THAT. The marker with ordinal value 2 occurs six times and the marker with ordinal value 3 occurs twice as do their corresponding argument substrings in the original text. There is no marker of ordinal value four since the scanning for the text already contained a segment gap in the middle of the string COINCIDE created by the previous argument substring IN.

The £(PF,) function asks for the print-out of the null-named form and since there is not such a form, for the moment, the TRAC processor responds with the appropriate message.

The null form is now defined and its content is the read-in string of the following two lines of input. The print form, £(PF,), prints out the null form as it was stored, since no change has occurred to it. The null form is segmented and its substrings OF and ABNORMALITIES are replaced by corresponding internal markers. The subsequent £(PF,) function prints out now the content of the form and the internal marker indicators.

A similar procedure consisting of the functions RS, DS, PF, SS and

PF is carried out on the next read-in string which is stored as AB. The following function £(PF AB) is coded in the wrong way and the appropriate message is given.

Finally, the form TEXT and the null form, both of which are already segmented, are segmented for a second time, some more segment gaps are created and internal markers replace argument substrings. The newly created markers and their correspondence to the arguments of the segment string function can be inspected in the print-out of these two forms caused by the two print form functions ££(PF,TEXT) and ££(PF,). These are presented here in their neutral mode to show that when the function is null there is no difference in execution between the active and the neutral modes.

IIC.7.2 Example Test-5 for function PF

The two first lines for the title of the output are obtained in the same way as in the previous example.

The second time the rs function of the idling procedure is replaced by the four following lines typed in from the typewriter. This input is evaluated from the active string. The nested rs and ds expression stores the string read THESE STARK CONDITIONS FOUND A BLEAK PARALLEL IN THE POLITICAL CLIMATE WHICH THEN PREVAILED BETWEEN COUNTRIES., and names it A1. A similar nested expression names as A2 the string read THESE ARRANGEMENTS(,) LIKE THE STEPS OF A FORMAL DANCE(,) CONVEY LITTLE OF THE FEELINGS OF THE PEOPLE CONCERNED. The print form functions £(PF,A1) and £(PF,A2) print out the strings A1 and A2. There are no segment gap indicators printed, since the forms have not been segmented.

The two following segment string functions create segment gaps which are indicated by internal markers. There are three internal markers for the form A1, one for each of the substrings SE STARK COND,

BLEAK PARALLEL and AL CLIMATE WH. The segmented form A1 is printed out by the £(PF,A1) function. The input string has been changed to the following:

THEe_1ITIONS FOUND A e_2 IN THE POLITICe_3ICH THEN PREVAILED BETWEEN COUNTRIES.

It can be checked easily that the segment gap indicators correspond to the replaced argument substrings. Three such indicators are printed, as might be expected.

There are four internal markers for the form A2, one for each of the substrings FORMAL DANCE, THESE, THE PEOPLE and OF. The segmented form A2 is printed out by the £(PF,A2) function. The corresponding input string has now been changed to the following:

e_2 ARRANGEMENTS, LIKE THE STEPS e_4 A e_1, CONVEY LITTLE e_4 THE FEELINGS e_4 e_3 CONCERNED.

The segment gap indicators correspond to the internal markers. There are three internal markers of ordinal value 4, since there were four occurrences of the substring OF. Thus the symbol e_4, which is the indicator for this internal marker, occurs three times in the above written form.

Two neutral call functions follow. Each one of them is the second argument of the idling procedure. It is replaced by its value which is printed out. It must be noted that the printed output is the same as the input, although the forms A1 and A2 have been segmented. This is accomplished by providing both call functions with arguments identical to the arguments of the corresponding segment string functions. Thus each internal marker which has replaced a substring of text is now replaced by the same substring, so the value of the function is the form as it was stored before segmentation occurred.

IIC.8 Group-6 Programs for Testing EQ Function

The equal function, £(EQ,S1,S2,T,F), provides the means for branching. The second and third arguments are tested for equality. Only when these two arguments contain the same characters are they considered equal. The fourth argument substring is the value of the function when the tested arguments are equal, otherwise the fifth argument is the value. The fourth and fifth arguments may be procedures. Thus control can be transferred to different procedures depending on the equality of the tested strings.

The execution of each program in this group is followed by a second one using the trace on function to make clear the execution from the neutral string.

IIC.8.1 Example Test-2 for function EQ

The first time the idling procedure is loaded its read string function is replaced by the six consecutive lines of input typed in from the typewriter. This input is evaluated from the active string. The first two print string functions cause the print-out of the title.

The nested read and define string expression £(DS,TEXT,££(RS)) causes the string NOT ALL THAT TEMPTS YOUR WANDERING EYES to be read in from the typewriter and stored in memory as a form named TEXT. The nested call and define string expression £(DS,TEMP,££(CL,TEXT)) creates another form TEMP with the same content as the form TEXT.

The expression £(SS,TEXT,£(RS)) causes the segmentation of the form TEXT. The argument substrings TEMPTS, WANDERING are provided by the read string function. The form TEXT is retrieved and internal markers replace the above substrings.

A procedure named PROC is defined by the following define string

function. This procedure is a loop since its last function, £(CL,PROC), is a call to itself. The procedure PROC contains the following parts:

(1) a define string function which is a redefinition of the form TEXT since the third argument is a macro call to the form TEXT and parameters are provided by the read string function;

(2) the equal function which tests the two forms, TEMP, which holds the original text, and TEXT, which is the redefined form.

The equal function provides a means of exit from the loop. When the tested forms are equal, the fourth argument is the value of the function and it does not contain the call to the procedure. The fifth argument is executed when the tested forms are not equal. This argument contains one print string function to print the form TEXT for inspection, a segment string function to replace some of its arguments by internal markers and finally an active call to the procedure PROC which causes another copy of this procedure to be processed from the active string.

The above procedure is put into action by an active call £(CL,PROC) and it is executed several times by providing dummy variables and parameters from the typewriter. The form TEXT is printed out every time the procedure is executed and it is seen that it is different from the original form TEMP. When the form TEXT becomes equal to the form TEMP the fourth argument of the equal function is executed. An END PROGRAM message is printed out and both forms TEXT and TEMP are printed, so their equality may be checked easily. There is no call to the procedure and execution terminates.

IIC.8.2 Example Test-8 for function EQ

The read string function of the idling procedure is replaced by four consecutive lines read in from the typewriter. This input character string is evaluated from the active string.

The first two print string functions cause the usual print-out of the two lines of output. The following define string function defines the procedure Q. This procedure contains the functions referred to below in the order in which they occur:-

(1) £(DS,S,X)

(2) £(SS,S,Y)

(3) £(DS,Z,££(CL,S,A))

(4) £(EQ,££(CL,S,Y),££(CL,Z),££(CL,Z),(£(CL,Q,££(CL,Z),Y)))

The segment string function that follows brings the procedure Q from storage, replaces all occurrences of X and Y with internal markers of ordinal value 1 and 2 and puts the segmented procedure back in store in the place of the old one. There is one marker of ordinal value 1 and three of ordinal value 2.

The second time the idling procedure is loaded, its read string function is replaced by the expression £(PS,£(CL,Q,ABCBCC,ABC)). The active call function is a macro call to the segmented procedure Q. The form Q is brought from storage, its internal markers are replaced by the strings ABCBCC and ABC and it is evaluated from the active string.

The define string function creates the form S with content ABCBCC. This form is segmented by the subsequent segment string operation and an internal marker replaces the string ABC. The nested call and define string function is now evaluated. The neutral call function replaces the internal marker of the form S with the string A. Since it is a neutral function, its value, which is ABCC, replaces it on the neutral string. A new form, Z, with content ABCC is now created by the define string function.

The equal function is now evaluated. The second argument is a neutral call and it is replaced by its value, the string ABCBCC. The third argument is another neutral call to the form Z and it is replaced

by the string ABCC. Since the second and third arguments of the function are not equal the value is the fifth argument which is an active call to the procedure Q. The active function £(CL,Q,££(CL,Z),ABC) replaces the equal function in the active string. This function is now evaluated and the argument ££(CL,Z) is replaced by its value, the string ABCC. The active call £(CL,Q,ABCC,ABC) now causes the retrieval of the procedure Q and its evaluation with parameters ABCC, ABC.

The define string function creates the form S with content ABCC. This form is segmented by the subsequent segment string operation and an internal marker replaces the string ABC. The nested call and define string function is now evaluated. The neutral call replaces the internal marker of the form S with the string A. Since it is a neutral function, its value, which is AC, replaces it on the neutral string. A new form, Z, with content AC is now created by the define string function.

The equal function is now evaluated. The second argument is a neutral call and it is replaced by its value, the string ABCC. The third argument is another neutral call to the form Z and it is replaced by the string AC. Since the second and third arguments of the function are not equal, the value is the fifth argument, which is an active call to the procedure Q. The active function £(CL,Q,££(CL,Z),ABC) replaces the equal function in the active string. This function is now evaluated and the argument ££(CL,Z) is replaced by its value, the string AC. The active call £(CL,Q,AC,ABC) now causes the retrieval of the procedure Q and its evaluation with parameters AC and ABC.

The define string function creates the form S with content AC. This form is not segmented since there is no substring ABC in its value. The nested call and define string function is now evaluated. There is no replacement in the call function since the form S is not segmented.

Thus the form Z is defined now with content AC, the same as the form S.

The equal function is now evaluated. The second argument is a neutral call and it is replaced by its value, which is the string S since the form S is not segmented. The third argument is a neutral call to the form Z which was defined previously and has the same content as the form S. Thus the second and third arguments are identical and the value of the equal function is the fourth argument. The fourth argument is the string AC which replaces the neutral call to the form Z. This string now becomes the second argument of the ps function of the idling procedure and it is printed out.

IIC.9 Listing of TRAC Applications

(See pages 196-223.)

IIC.10 Execution of TRAC Applications

(See pages 224-309.)

IIC.9 Listing of TRAC Applications

. . . pages 196-223

```
                                TRAC  APPLICATIONS                                    1
*****************************************************************************       2
                                                                                    3
          GROUP-1  PROGRAMS FOR TESTING RS,PS,CL,DS,SS FUNCTIONS                    4
                                                                                    5
*****************************************************************************       6
££(PS,-------------------------------------------------------------------------)     7
££(PS,***************TEST-1 FUNCTIONS  RS*PS*CL*DS*SS**************************)     8
£(DS,A1,£(RS))£(SS,A1,M1,M2,M3)£(DS,A2,(£(CL,A1,MOUNTAINS,FOUNTAINS,GRAVES)))£(P    9
S,(£(CL,A2)))£(PS,££(CL,A2))£(PS,£(CL,A2))£(PS,-------------------------------     10
-----------------------------------------)'                                        11
OVER THE M1 AND OVER THE WAVES((,))UNDER THE M2 AND UNDER THE M3'                   12
'                                                                                   13
*****************************************************************************       14
££(PS,-------------------------------------------------------------------------)     15
££(PS,***************TEST-2 FUNCTIONS  RS*PS*CL*DS*SS**************************)     16
£(DS,XYZ,££(RS))£(DS,XYW,££(RS))££(SS,XYZ,PLEASURE,INFANT)££(SS,XYW,DART,PRAISE,    17
REST)'                                                                              18
UNDER INFANT THAT ARE DEEPEST(,)WHICH PLEASURE OBEY'                                19
OVER REST THAT ARE DART LOVE WILL FIND OUT THE PRAISE'                              20
£(CL,XYZ,NEPTUNE,FLOODS)'                                                           21
£(CL,XYW,STEEPEST,WAY,ROCKS)'                                                       22
£(PS,£(CL,XYZ)*****£(CL,XYW))£(PS,---------------------------------------------     23
---------------------------------)'                                                24
'                                                                                   25
*****************************************************************************       26
££(PS,-------------------------------------------------------------------------)     27
££(PS,***************TEST-3 FUNCTIONS  RS*PS*CL*DS*SS**************************)     28
£(DS,POEM,(£(RS)))'                                                                 29
(£(CL,POEM))'                                                                       30
££(CL,POEM)'                                                                        31
£(CL,POEM)'                                                                         32
WHEN I THE DAWN USED TO ADMIRE((,))AND PRAISED THE COMING DAY'                      33
£(DS,POEM,£(CL,POEM))'                                                              34
WHEN I THE DAWN USED TO ADMIRE((,))AND PRAISED THE COMING DAY'                      35
£(SS,POEM,DAWN,PRAISED,ADMIRE)'                                                     36
££(CL,POEM)'                                                                        37
£(CL,POEM,1,2,3)'                                                                   38
```

```
£(CL,POEM,MARKER1,MARKER2,MARKER3)'                                                    39
£(CL,POEM,DAWN,PRAISED,ADMIRE)'                                                        40
£(RS)'                                                                                 41
£(PS,----------)'                                                                      42
'                                                                                      43
**********                                                                             44
££(PS,----------)                                                                      45
££(PS,**********TEST-4 FUNCTIONS  RS*PS*CL*DS*SS**********)                            46
£(DS,TEXT,££(RS))£(SS,£(RS))£(PS,TEXT**£(CL,TEXT))£(PS,TEXT WITH MARKERS**£(CL,T      47
EXT,1,2))£(DS,NEW-TEXT,£(CL,TEXT,THOUGHT,MUST TAKE))£(PS,NEW-TEXT**£(CL,NEW-TEXT      48
))'                                                                                    49
I LITTLE THOUGHT THE GROWING FIRE MUST TAKE MY REST AWAY'                              50
TEXT,THOUGHT,MUST TAKE'                                                                51
£(RS)'                                                                                 52
£(PS,----------)'                                                                      53
'                                                                                      54
**********                                                                             55
££(PS,----------)                                                                      56
££(PS,**********TEST-5 FUNCTIONS  RS*PS*CL*DS*SS**********)                            57
£(DS,V1,££(RS))£(DS,V2,££(RS))£(DS,V3,££(RS))££(SS,V1,MM,NN,LL)£(SS,V3,1,2)£(PS,      58
£(CL,V1,CHARMS,CHILDHOOD,IN THE)***£(CL,V2))£(PS,£(CL,V3,AGE FROM,MORE AWAY))'         59
YOUR     MM IN HARMLESS       NN  LAY LIKE METALS LL     MINE'                         60
YOUR CHARMS IN HARMLESS CHILDHOOD LAY LIKE METALS IN THE MINE'                         61
1 NO FACE TOOK 2'                                                                      62
£(RS)'                                                                                 63
£(PS,----------)'                                                                      64
'                                                                                      65
**********                                                                             66
££(PS,----------)                                                                      67
££(PS,**********TEST-6 FUNCTIONS  RS*PS*CL*DS*SS**********)                            68
£(DS,X,£(RS))£(PS,£(CL,N,£(DS,N,££(CL,X)£(CL,X)££(CL,X))))£(PS,X****££(CL,X))£(P      69
S,N*****££(CL,N))£(PS,----------                                                       70
----------)'                                                                          71
((£(DS,X,YOU COMMON PEOPLE OF THE SKIES(,)WHAT ARE YOU(,)WHEN THE MOON SHALL RIS      72
E)))'                                                                                  73
'                                                                                      74
**********                                                                             75
£(PS,----------)                                                                       76
```

```
££(PS,***************TEST-7 FUNCTIONS  RS*PS*CL*DS*SS*************************)
£(DS,XY,££(RS))£(SS,XY,M1,M2,M3)££(PS,XY*****££(CL,XY,WORDS,JUDGE YOU TO,VIRTUES
,DEMAND,DESPAIR))£(DS,XZ,££(RS))£(SS,XZ,1,2,3,4)££(PS,XZ*****££(CL,XZ,LIFE,LOVE,
HEART,EYES,CHANGE,COMMAND))£(PS,-----------------------------------------------
--------------------------------)'
SO WELL YOUR M1 HIS NOBLE M3 PRAISE,THAT ALL BOTH M2  RELATE THEM TRUE'
THOU ART MY 1,MY 2,MY 3,THE VERY 4 OF ME'
'
**************************************************************************************
£(PS,-----------------------------------------------------------------------------)
££(PS,***************TEST-8  FUNCTIONS  RS*PS*CL*DS*SS*************************)
£(DS,XY,££(RS))£(SS,XY,M1,M2,M3,M4,M5)££(PS,XY*****£(CL,XY,TH,OU,AND))££(DS,XZ,£
(RS))£(SS,XZ,M1,M2,M3)££(PS,XZ*****£(CL,XZ,LS,PA,ALL TIME AND))£(PS,XY**£(CL,XY)
   XZ***£(CL,XZ))£(PS,-----------------------------------------------------------
----------------------)'
M1M2GH M5 SEAS M3 LM3 BETWIXT M4 US BOM1(,) M2R FAIM1 M3 TROM1'
LIKE SEM2RATED SOUM1((,))M3 SM2CE CONTROM1'
'
**************************************************************************************
£(PS,-----------------------------------------------------------------------------)
££(PS,***************TEST-9  FUNCTIONS  RS*PS*CL*DS*SS*************************)
£(DS,,££(RS))££(SS,,M1,M2,M3,M4,M5)£(PS,NULL NAME FORM********££(CL,,THOU CANST,
HUMBLE,SLEEP))££(DS,ZZZ,££(RS))£(SS,ZZZ,M1,M2,M3)£(PS,ZZZ***££(CL,ZZZ,IN HER,NOR
 GENTLE,SOUNDS THAT))£(PS,-----------------------------------------------------
-------------------------)'
M3,M3 AGAIN,MY M4 LYRE M5 FOR M1 NEVER M5 TELL MY M2 TALE'
IN M3 WILL PREVAIL,M2 THOUGHTS M1 INSPIRE'
'
**************************************************************************************
£(PS,-----------------------------------------------------------------------------)
££(PS,***************TEST-10 FUNCTIONS  RS*PS*CL*DS*SS*************************)
£(DS,QQQQ,£(RS))£(CL,QQQQ)£(PS,QQQQ****££(CL,QQQQ))'
(£(PS,HENCE(,)ALL YOU 2(,)AS SHORT AS ARE 1))'
£(SS,QQQQ,£(RS))£(DS,QQQQ,££(CL,QQQQ,£(RS)))£(CL,QQQQ)£(PS,QQQQ****££(CL,QQQQ))'
1,2'
M1,M2'
£(SS,QQQQ,£(RS))£(DS,QQQQ,££(CL,QQQQ,£(RS)))£(CL,QQQQ)£(PS,QQQQ****££(CL,QQQQ))'
M1,M2'
```

```
MARKER1,MARKER2'                                                                                                    115
£(SS,QQQQ,£(RS))£(DS,QQQQ,££(CL,QQQQ,£(RS)))£(CL,QQQQ)£(PS,QQQQ****££(CL,QQQQ))'                                    116
MARKER1,MARKER2'                                                                                                    117
THE NIGHTS,VAIN DELIGHTS'                                                                                           118
£(PS,----------)'                                                                                                   119
'                                                                                                                   120
**********                                                                                                          121
£(PS,----------)                                                                                                    122
££(PS,**********TEST-11 FUNCTIONS  RS*PS*CL*DS*SS**********)                                                        123
£(DS,QQQQ,(£(PS,HENCE(,)ALL YOU 2(,)AS SHORT AS ARE 1)£(PS,QQQQ****££(CL,QQQQ))£                                    124
(SS,QQQQ,£(RS))£(DS,QQQQ,££(CL,QQQQ,£(RS)))£(CL,QQQQ)))£(CL,QQQQ)''                                                 125
1,2'                                                                                                                126
M1,M2'                                                                                                              127
M1,M2'                                                                                                              128
MARKER1,MARKER2'                                                                                                    129
MARKER1,MARKER2'                                                                                                    130
THE NIGHTS,VAIN DELIGHTS'                                                                                           131
£(PS,----------)'                                                                                                   132
'                                                                                                                   133
**********                                                                                                          134
£(PS,----------)                                                                                                    135
££(PS,**********TEST-12 FUNCTIONS  RS*PS*CL*DS*SS**********)                                                        136
£(DS,ABC,££(RS))£(PS,ABC*****££(CL,ABC))££(SS,ABC,ABC1,ABC2,ABC3)£(PS,ABC MARKER                                    137
S SUBSTITUTED BY NULL*****££(CL,ABC))£(DS,PROCEDURE,(£(PS,£(CL,£(RS)))))£(PS,PRO                                    138
CEDURE*****££(CL,PROCEDURE))£(CL,PROCEDURE)£(CL,PROCEDURE)£(PS,ABC LAST CALL****                                    139
***££(CL,ABC))'                                                                                                     140
WELCOME(,)ABC3 ARMS AND ABC1(,)A SIGH THAT ABC2 MORTIFIES'                                                          141
ABC,1,2,3'                                                                                                          142
ABC,FIXED EYES,PIERCING,FOLDED'                                                                                     143
£(PS,----------)'                                                                                                   144
'                                                                                                                   145
**********                                                                                                          146
£(PS,----------)                                                                                                    147
££(PS,**********TEST-13 FUNCTIONS  RS*PS*CL*DS*SS**********)                                                        148
£(DS,XYZ,£(RS))£(PS,£(CL,XYZ))'                                                                                     149
THIS IS A CAT'                                                                                                      150
£(SS,XYZ,CAT)'                                                                                                      151
£(CL,XYZ,DOG)'                                                                                                      152
```

```
£(CL,XYZ,MOUSE)'                                                                153
£(CL,XYZ,SHEEP)'                                                                154
£(CL,XYZ,COW)'                                                                  155
£(CL,XYZ,HORSE)'                                                                156
£(CL,XYZ)'                                                                      157
£(DS,£(RS),££(CL,XYZ,CAT))£(PS,£(CL,ZYX))'                                      158
ZYX'                                                                            159
£(SS,ZYX,CAT)'                                                                  160
£(CL,ZYX,DOG)'                                                                  161
£(CL,ZYX,MOUSE)'                                                                162
£(CL,ZYX,SHEEP)'                                                                163
£(CL,ZYX,COW)'                                                                  164
£(CL,ZYX,HORSE)'                                                                165
£(CL,ZYX)'                                                                      166
£(PS,----------------------------------------------------------------------)'  167
'                                                                               168
**********************************************************************          169
£(PS,----------------------------------------------------------------------)    170
££(PS,**************TEST-14 FUNCTIONS  RS*PS*CL*DS*SS**************************)  171
£(DS,POEM,(£(DS,VERSEM,Y)))£(PS,POEM*****££(CL,POEM))££(SS,POEM,M,Y))£(PS,POEM**  172
***££(CL,POEM))£(PS,££(CL,£(RS)))£(CL,£(RS))£(PS,££(CL,VERSE1))'                173
POEM,1,TO ME ONE SAVAGE HUNTING SCENE((,))MY SOLE DELIGHT THE HEADLONG RACE''   174
POEM,1,TO ME ONE SAVAGE HUNTING SCENE((,))MY SOLE DELIGHT THE HEADLONG RACE''   175
£(PS,££(CL,£(RS)))£(CL,£(RS))£(PS,££(CL,VERSE2))'                               176
POEM,2,AND FRANTIC HURRY OF THE CHASE TO START((,))PURSUE((,))AND BRING TO BAY'  177
POEM,2,AND FRANTIC HURRY OF THE CHASE TO START((,))PURSUE((,))AND BRING TO BAY'  178
£(PS,----------------------------------------------------------------------)'  179
'                                                                               180
**********************************************************************          181
£(PS,----------------------------------------------------------------------)    182
££(PS,**************TEST-15 FUNCTIONS  RS*PS*CL*DS*SS**************************)  183
£(DS,VERSE1,HE GAVE US THIS ETERNAL SPRING WHICH HERE ENAMELS EVERYTHING(,))'    184
££(CL,VERSE1)'                                                                  185
£(DS,VERSE1,HE GAVE US THIS ETERNAL SPRING WHICH HERE ENAMELS EVERYTHING(,))££(C  186
L,VERSE1)'                                                                      187
£(DS,£(PS,GIVE NAME FOR TEXT)£(RS),£(PS,GIVE THE TEXT)£(RS))££(CL,£(PS,GIVE THE  188
NAME OF TEXT YOU CALL)£(RS))'                                                   189
VERSE1'                                                                         190
```

```
HE GAVE US THIS ETERNAL SPRING WHICH HERE ENAMELS EVERYTHING(,)'                                                    191
VERSE1'                                                                                                             192
£(DS,PROGRAM TEXT STORE,(£(DS,£(PS,GIVE NAME FOR TEXT)£(RS),£(PS,GIVE THE TEXT)£                                   193
(RS))£(PS,£(CL,£(PS,GIVE THE NAME OF TEXT YOU CALL)£(RS)))£(CL,PROGRAM TEXT STOR                                   194
E)))£(CL,PROGRAM TEXT STORE)'                                                                                       195
VERSE2'                                                                                                             196
AND SENDS THE FOWLS TO US IN CARE ON DAILY VISITS THROUGH THE AIR'                                                  197
VERSE2'                                                                                                             198
VERSE3'                                                                                                             199
HE HANGS IN SHADES THE ORANGE BRIGHT LIKE GOLDEN LAMPS IN A GREEN NIGHT'                                            200
VERSE3'                                                                                                             201
£(PS,VERSE1**££(CL,VERSE1)    VERSE2**££(CL,VERSE2)    VERSE3**££(CL,VERSE3))'                                      202
£(PS,----------)'                                                                                                   203
'                                                                                                                   204
**********                                                                                                          205
£(PS,----------)                                                                                                    206
££(PS,**********TEST-16 FUNCTIONS  RS*PS*CL*DS*SS**********)                                                        207
£(DS,EX,PROGRAM SHOWING THE USE OF A PROCEDURE)'                                                                    208
£(DS,VERSE1,THEN LET NOT WHAT I CANNOT HAVE)'                                                                       209
£(DS,VERSE2,MY CHEER OF MIND DESTROY)'                                                                              210
£(DS,VERSE3,WILST THUS I SING((,))I AM A KING((,))ALTHOUGH A POOR BLIND BOY)'                                       211
£(DS,PROCEDURE,(££(PS,(GIVE NAME**********))£(PS,£(CL,£(RS)))£(CL,PROCEDURE)))'                                    212
£(CL,PROCEDURE)'                                                                                                    213
EX'                                                                                                                 214
VERSE1'                                                                                                             215
VERSE2'                                                                                                             216
VERSE3'                                                                                                             217
£(PS,----------)'                                                                                                   218
'                                                                                                                   219
**********                                                                                                          220
£(PS,----------)                                                                                                    221
££(PS,**********TEST-17 FUNCTIONS  RS*PS*CL*DS*SS**********)                                                        222
£(RS)£(RS)£(RS)£(RS)£(RS)£(RS)£(D£(PS,THE DEFINITION OF FORMS BEGINS)S,£(RS))                                       223
£(DS,,ABC IS MY ABDD) £(SS,,ABC,ABDD)          £(DS,£(RS))£(DS,£(RS))£(DS,£(RS                                      224
                                    ))££(SS,X2,MARKER1,MARKER2,MARKER3)££(SS,X4,PARI                                225
S,LEARNING,FRENCH)££(SS,X1,SMITH,SEAT,HOUSE,COMMONS)£(RS)£(RS)£(RS)£(RS)£(RS)£(R                                   226
S)£(SS,X3,MARKER1,MARKER2,MARKER3,MARKER4)£(RS)£(RS)£(PS,£(CL,))£(PS,£(CL,,))£(P                                   227
S,£(CL,,THIS,BOOK))'                                                                                                228
```

```
£(PS,                    STORE  OF  FORMS)'                                                    229
£(PS,*************************************************************************)'            230
£(PS,              X1  MR SMITH HAS A SEAT IN THE HOUSE OF COMMONS)'                           231
£(PS,              X2  THE CLASSIC ROUTE TO THE MIDNIGHT SUN)'                                 232
£(PS,              X3  AFTER LUNCH YOU CONTINUE BY TRAIN A ROOM IS RESERVED FOR Y              233
OU)'                                                                                           234
£(PS,              X4  HE WENT TO PARIS WITH THE INTENTION OF LEARNING FRENCH)'                235
X1,MR SMITH HAS A SEAT IN THE HOUSE OF COMMONS'                                                236
X2,MARKER1 CLASSIC MARKER2 TO MARKER1 MARKER3'                                                 237
X3,AFTER LUNCH MARKER1 CONTINUE MARKER3 MARKER4 MARKER2 A ROOM IS RESERVED FOR M               238
ARKER1'                                                                                        239
X4,HE WENT TO PARIS WITH THE INTENTION OF LEARNING FRENCH'                                     240
£(PS,£(CLX1,TAYLOR,FLAT,CITY,LONDON)£(CL,X1,TAYLOR,FLAT,CITY,LONDON))'                        241
£(PS,£(CL,X2))'                                                                                242
£(PS,£(CL,X2,THE,ROUTE,MIDNIGHT SUN))'                                                        243
£(PS,£(CL,X2,))'                                                                               244
£(PS,£(CL,X3,YOU,BY TRAIN))'                                                                   245
£(PS,£(CL,X5))'                                                                                246
£(PS,£(CL,X3,YOU,BY TRAIN))'                                                                   247
£(PS,£(CL,X4,STOCKHOLM,MEETING,FRIENDS,TELEIE,COSTA))'                                         248
£(PS,-------------------------------------------------------------------)'                    249
'                                                                                              250
*************************************************************************************        251
£(PS,-------------------------------------------------------------------------------)        252
££(PS,***************TEST-18 FUNCTIONS  RS*PS*CL*DS*SS*****************************)        253
£(DS,V1,£(RS))£(SS,V1,1,2,3)'                                                                  254
WITH HEAVY 3 I OFTEN 1 MOURN MY 2 WOE'                                                         255
£(DS,PROG,(£(PS,£(CL,£(RS)))£(CL,PROG)))'                                                      256
£(CL,PROG)'                                                                                    257
V1,M1,M2,M3'                                                                                   258
V1,MM1,MM2,MM3'                                                                                259
V1,MARKER1,MARKER2,MARKER3'                                                                    260
V1,HEAR YOU,HAPLESS,SIGHS'                                                                     261
£(PS,-------------------------------------------------------------------------------)'        262
'                                                                                              263
*************************************************************************************        264
£(PS,-------------------------------------------------------------------------------)        265
££(PS,***************TEST-19 FUNCTIONS  RS*PS*CL*DS*SS*****************************)        266
```

```
£(DS,PROG,(£(PS,££(CL,£(RS)))£(CL,PROG)))'                                      267
£(DS,RECORD,(£(DS,X,Y)))£(PS,££(CL,RECORD))£(SS,RECORD,X,Y)£(PS,££(CL,RECORD))' 268
£(CL,PROG)'                                                                     269
RECORD,VERSE1,THE FOND COMPLAINT((,))MY SONG((,))DISPROVE'                      270
£(PS,££(CL,VERSE1))£(PS,££(CL,RECORD,VVV,WWW))'                                 271
RECORD,VERSE2,AND JUSTIFY THE LAWS OF JOVE'                                     272
£(PS,££(CL,VERSE2))£(PS,----------------------------------------------------- 273
----------------------)'                                                        274
'                                                                               275
*************************************************************************** 276
                                                                                277
          GROUP-2  PROGRAMS FOR TESTING TN,TF FUNCTIONS                         278
                                                                                279
*************************************************************************** 280
££(PS,-------------------------------------------------------------------) 281
££(PS,*******************1 TEST FOR FUNCTIONS TN(,)TF ************************) 282
£(TN)'                                                                          283
£(DS,1,(£(PS,POOR SOUL(,)THE CENTRE OF MY SINFUL EARTH)£(SS,1,£(RS))£(DS,1,££(CL 284
,1,£(RS)))))£(CL,1)£(CL,1)'                                                     285
SOUL,EARTH'                                                                     286
EARTH,SOUL'                                                                     287
£(TF)'                                                                          288
£(PS,-------------------------------------------------------------------)'     289
'                                                                               290
*************************************************************************** 291
££(PS,-------------------------------------------------------------------) 292
££(PS,*******************2 TEST FOR FUNCTIONS TN(,)TF************************) 293
£(TN)'                                                                          294
£(SS,BUILDLINE,1,£(DS,BUILDLINE,1 AND HIS DOG))                                 295
£(DS,THISTHING,ONE MAN)                                                         296
£(SS,WTM,1,£(DS,WTM,1 WENT TO MOW (,)))                                         297
£(TF)'                                                                          298
£(DS,WTMAM,WENT TO MOW A MEADOW.)                                               299
£(PS,££(CL,BUILDLINE))£(PS,££(CL,THISTHING))£(PS,££(CL,WTM))£(PS,££(CL,WTMAM))' 300
££(PS,-------------------------------------------------------------------)'    301
'                                                                               302
*************************************************************************** 303
££(PS,-------------------------------------------------------------------) 304
```

```
&&(PS,**********3 TEST FOR FUNCTIONS TN(,)TF**********)
&(TN)&(DS,,WHOSE M1 NOT HIS M2 ARE)&(SS,,M1,M2)&(DS,&(RS))&(DS,&(RS))&(DS,&(RS))
&(DS,&(RS))&&(SS,X2,MARKER1,MARKER2,MARKER3)&&(SS,X4,1MM,2MM,3MM)&&(SS,X1,1,2,3)
&(PS,&(CL,))&(PS,&(CL,X1,O HOW I,TRAVEL,TREAD AGAIN))&(PS,&(CL,X2,THAT I MIGHT O
NCE MORE,WHERE FIRST,GLORIOUS TRAIN))&(PS,&(CL,X3,MY SOUL,TOO MUCH))&(PS,&(CL,X4
&(SS,X3,MARKER1,MARKER2,MARKER3,MARKER4)&(PS,&(CL,X3,MY SOUL,TOO MUCH)),AND WHEN
,TO THE URN,STATE I COME))&(PS,&(CL,,PASSIONS,MASTERS))&(TF)&(PS,&(CL,)***&(CL,X
1)***&(CL,X2)***&(CL,X3)***&(CL,X4))&&(PS,----------
----------)'
X1,1 LONG TO 2 BACK((,))AND 3 THAT ANCIENT TRACK'
X2,MARKER1 REACH THAT PLAIN((,)) MARKER2 I LEFT MY MARKER3'
X3,BUT AH MARKER1 MARKER3 WITH MARKER2 STAY MARKER4'
X4,1MM THIS DUST FALLS 2MM((,))IN THAT 3MM((,))RETURN'
'
**********
&&(PS,----------)
&&(PS,**********4 TEST FOR FUNCTIONS TN(,)TF**********)
&(DS,,&&(RS))&(TN)&(SS,,MARKER1,MARKER2,MARKER3,MARKER4,MARKER5)&&(PS,&(CL,,EAR,
IN,FORWARD))&&(DS,XXX,&(RS))&&(SS,XXX,1,2,3,4)&(TF),&(PS,&(CL,XXX,TIS,TO,THE,IN)
)&(PS,----------
-----)''
THE MARKER3 MARKER4YOUTH THAT WOULD APPMARKER1(,)MUST MARKER5 NOW FORSAKE HIS  M
ARKER4MUSES DMARKER1(,) MARKER5 NOR MARKER2 THE SHADOWS SMARKER2G MARKER4 MARKER
5HIS  NUMBERSMARKER4 LANGUISHMARKER2G.'
1 TIME 2 LEAVE 3 BOOKS 4 DUST'
'
**********
&&(PS,----------)
&&(PS,**********5 TEST FOR FUNCTIONS TN(,)TF **********)
&(DS,CC,&(RS))&&(TN)&&(DS,DD,&&(RS))&(SS,CC,1,2,3,4,5)&(SS,DD,6,7,8,9)&(TF)&(PS,
FIRST VERSE  **&(CL,CC,A,B,C,D,E)        SECOND VERSE **&(CL,DD,F,G,H,I))&(RS)'
MORTILITY((,))25HOL4 1N4 F51R WH1T 1 3H1NG5 OF FL5SH IS H5R5'
T89NK 8OW MANY ROYAL BONES SLEEP W9T89N T8ESE 8EARS 06 STONES'
&(PS,----------)'
'
**********
&&(PS,----------)
&&(PS,**********6 TEST FOR TN(,)TF  FUNCTIONS**********)
```

```
£(DS,AA,£(RS)£(TN))££(SS,AA,1A,1B,1C)£(DS,PROG,(£(PS,£(CL,£(RS))))£(TF)£(CL,PRO
G)£(PS,-------------------------------------------------------------------
------)'
1A((,))OR 1B((,))OR 1C IN ARMS((,))  WHOSE CHANCE ON THESE DEFENCELESS DOORS MAY
 SEIZE'
AA,CAPTAIN,COLONEL,KNIGHT'
'
*************************************************************************
££(PS,-------------------------------------------------------------------)
££(PS,*******************7 TEST FOR TN(,)TF  FUNCTIONS*********************)
£(DS,ABAB,£(RS)££(TN))£(SS,ABAB,GREEK),£(DS,BABA,(£(CL,ABAB,ATHENIAN))),£(PS,(£(
CL,BABA))),£(PS,££(CL,BABA)),£(PS,£(CL,BABA)££(TF))£(PS,-----------------------
-------------------------------------------------)'
TO SAVE GREEK WALLS FROM RUIN BARE.'
'
*************************************************************************
££(PS,-------------------------------------------------------------------)
££(PS,*******************8 TEST FOR FUNCTIONS TN(,)TF *********************)
£(DS,SONG,£(RS)£(TN)),
                £(SS,SONG,SOME,THING,ELSE),£(PS,£(CL,SONG,ONE,MAN,ONE MAN)),£(P
S,££(CL,SONG£(TF),TWO,MEN,(TWO MEN   ONE MAN)))£(RS)'
SOME THING WENT TO MOW WENT TO MOW A MEADOW ELSE AND HIS DOG WENT TO MOW A MEADO
W'
£(PS,-------------------------------------------------------------------)'
'
*************************************************************************

            GROUP-3  PROGRAMS FOR TESTING DA FUNCTION

*************************************************************************
££(PS,-------------------------------------------------------------------)
££(PS,*******************1  TEST FOR FUNCTION DA **************************)
£(DS,SONG,£(RS)),£(SS,SONG,SOME,THING,ELSE),£(PS,£(CL,SONG,ONE,MAN,ONE MAN)),£(P
S,£(DA)££(CL,SONG,TWO,MEN,(TWO MEN   ONE MAN)))'
SOME THING WENT TO MOW WENT TO MOW A MEADOW ELSE AND HIS DOG WENT TO MOW A MEADO
W'
'
*************************************************************************
```

```
££(PS,----------)                                                                  381
££(PS,**********2  TEST FOR FUNCTION DA **********)                                382
£(DS,A,£(RS)),£(SS,A,ALIVE),£(DS,B,(£(CL,A,DEAD))),£(PS,(£(CL,B))),£(PS,££(CL,B)  383
)£(PS,£(CL,B))£(LN,NAME  )£(DA)£(LN,NAME  )£(PS,£(CL,B))'                          384
NO LONGER MOURN FOR ME WHEN I AM ALIVE'                                            385
'                                                                                  386
**********                                                                         387
££(PS,----------)                                                                  388
££(PS,**********3  TEST FOR FUNCTION DA **********)                                389
£(DS,VERSE1,£(RS)),£(DS,VERSE2,££(RS)),££(SS,VERSE1,1,2,3),££(SS,VERSE2,1,2,3,4)  390
£(PS,£(CL,VERSE1,LEST,WISE WORLD,MOAN))£(PS,£(CL,VERSE2,MOCK,WITH,AFTER,GONE))£(  391
LN,*****)£(DA)£(LN,*****)£(PS,£(CL,VERSE1))£(PS,£(CL,VERSE2))'                     392
1 THE 2 SHOULD LOOK INTO YOUR 3'                                                   393
AND I YOU 2 ME 3 I AM 4.'                                                          394
'                                                                                  395
**********                                                                         396
££(PS,----------)                                                                  397
££(PS,**********4  TEST FOR FUNCTION DA **********)                                398
£(DS,A1,£(RS)),££(SS,A1,MARKER1,MARKER2),£(DS,PROG,(£(PS,£(CL,£(RS))))),£(CL,PRO  399
G)£(LN,*****)£(£(DA)CL,PROG)'                                                      400
THROUGH MARKER2 OF MY LONG MARKER1 STAY'                                           401
A1,FRUITLESS,DISCONTENT'                                                           402
'                                                                                  403
**********                                                                         404
££(PS,----------)                                                                  405
££(PS,**********5  TEST FOR FUNCTION DA **********)                                406
££(DS,,££(RS)),£(SS,,11,22,33,44,55)££(PS,£(CL,,SWANS,ALONG,SOFTLY,GOODLY,THAT))  407
££(DS,XZ,£(RS)),££(SS,XZ,MARKERS)£(PS,£(CL,XZ,FAIRER BIRDS))£(LN,*****)£(DA)£(PS  408
,£(CL,))£(PS,£(CL,XZ))'W                                                           409
WITH 55 I SAW TWO 11 OF 44 HUE(,)COME 33 SWIMMING DOWN 22 THE LEE'                 410
TWO MARKERS I YET DID NEVER SEE'                                                   411
'                                                                                  412
**********                                                                         413
££(PS,----------)                                                                  414
££(PS,**********6  TEST FOR FUNCTION DA **********)                                415
£(SS,BUILDLINE,1,£(DS,BUILDLINE,1 AND HIS DOG))                                    416
£(DS,THISTHING,ONE MAN)                                                            417
£(SS,WTM,1,£(DS,WTM,1 WENT TO MOW (,)))                                            418
```

```
£(DS,WTMAM,WENT TO MOW A MEADOW.)                                                                    419
£(PS,££(CL,BUILDLINE))£(PS,££(CL,THISTHING))£(PS,££(CL,WTM))£(PS,££(CL,WTMAM))£(                    420
LN,*****)£(DA)£(LN,*****)£(PS,£(CL,BUILDLINE))'                                                     421
'                                                                                                    422
*****************************************************************************************          423
££(PS,-------------------------------------------------------------------------)                    424
££(PS,*******************7  TEST FOR FUNCTION DA ******************************)                    425
£(DS,QQ,(£(PS,(LET ENDLESS PEACE YOUR STEADFAST HEARTS ACCORD))£(SS,QQ,£(RS))£(D                     426
S,QQ,££(CL,QQ,£(RS)))))£(CL,QQ)£(CL,QQ)£(LN,*****)£(DA)£(CL,QQ)'                                     427
PEACE'                                                                                               428
WAR'                                                                                                 429
WAR'                                                                                                 430
SUMMER'                                                                                              431
'                                                                                                    432
*****************************************************************************************          433
££(PS,-------------------------------------------------------------------------)                    434
££(PS,*******************8  TEST FOR FUNCTION DA ******************************)                    435
£(RS)£(RS)£(RS)£(RS)£(RS)£(RS)£(D£(PS,THE DEFINITION OF FORMS BEGINS)S,£(RS))                        436
£(DS,,ABC IS MY ABDD) £(SS,,ABC,ABDD)          £(DS,£(RS))£(DS,£(RS))£(DS,£(RS                       437
                              ))££(SS,X2,MARKER1,MARKER2,MARKER3)££(SS,X4,PARI                       438
S,LEARNING,FRENCH)££(SS,X1,SMITH,SEAT,HOUSE,COMMONS)£(RS)£(RS)£(RS)£(RS)£(RS)£(R                     439
S)£(SS,X3,MARKER1,MARKER2,MARKER3,MARKER4)£(RS)£(RS)£(PS,£(CL,))£(PS,£(CL,,))£(P                     440
S,£(CL,,THIS,BOOK))'                                                                                 441
£(PS,                    STORE  OF  FORMS)'                                                          442
£(PS,*************************************************************************)'                    443
£(PS,            X1  MR SMITH HAS A SEAT IN THE HOUSE OF COMMONS)'                                   444
£(PS,            X2  THE CLASSIC ROUTE TO THE MIDNIGHT SUN)'                                         445
£(PS,            X3  AFTER LUNCH YOU CONTINUE BY TRAIN A ROOM IS RESERVED FOR Y                      446
OU)'                                                                                                 447
£(PS,            X4  HE WENT TO PARIS WITH THE INTENTION OF LEARNING FRENCH)'                        448
X1,MR SMITH HAS A SEAT IN THE HOUSE OF COMMONS'                                                      449
X2,MARKER1 CLASSIC MARKER2 TO MARKER1 MARKER3'                                                       450
X3,AFTER LUNCH MARKER1 CONTINUE MARKER3 MARKER4 MARKER2 A ROOM IS RESERVED FOR M                     451
ARKER1'                                                                                              452
X4,HE WENT TO PARIS WITH THE INTENTION OF LEARNING FRENCH'                                           453
£(PS,£(CLX1,TAYLOR,FLAT,CITY,LONDON)£(CL,X1,TAYLOR,FLAT,CITY,LONDON))'                               454
£(PS,£(CL,X2))'                                                                                      455
£(PS,£(CL,X2,THE,ROUTE,MIDNIGHT SUN))'                                                              456
```

```
£(PS,£(CL,X2,))'                                                                    457
£(PS,£(CL,X3,YOU,BY TRAIN))'                                                        458
£(PS,£(CL,X5))'                                                                     459
£(PS,£(CL,X3,YOU,BY TRAIN))'                                                        460
£(PS,£(CL,X4,STOCKHOLM,MEETING,FRIENDS,TELEIE,COSTA))'                              461
£(LN,*****)££(DA)£(LN,*****)'                                                       462
'                                                                                   463
**************************************************************************        464
                                                                                    465
          GROUP-4  PROGRAMS FOR TESTING LN FUNCTION                                 466
                                                                                    467
**************************************************************************        468
££(PS,----------------------------------------------------------------------)     469
£(PS,*******************TEST-1 LN FUNCTION*************************************)'  470
£(LN,NAME)££(LN,NAME  )£(LN COMMA MISSING)£(LN,)'                                   471
£(DS,VVV,(£(PS,(YOUTH IS FULL OF TROUBLES AGE IS FULL OF CARE))£(SS,VVV,£(RS))£(   472
DS,VVV,££(CL,VVV,£(RS)))))£(CL,VVV)£(CL,VVV)'                                       473
TROUBLES'                                                                           474
PLEASANCE'                                                                          475
£(LN,NAME)££(LN,NAME  )£(LN COMMA MISSING)£(LN,)'                                   476
££(PS,----------------------------------------------------------------)'          477
'                                                                                   478
**************************************************************************        479
££(PS,----------------------------------------------------------------------)     480
£(PS,*******************TEST-2 LN FUNCTION*************************************)'  481
£(SS,BUILDLINE,1,£(DS,BUILDLINE,1 AND HIS DOG))                                     482
£(DS,THISTHING,ONE MAN)                                                             483
£(SS,WTM,1,£(DS,WTM,1 WENT TO MOW (,)))                                             484
£(DS,WTMAM,WENT TO MOW A MEADOW.)                                                   485
£(PS,££(CL,BUILDLINE))£(PS,££(CL,THISTHING))£(PS,££(CL,WTM))£(PS,££(CL,WTMAM))'     486
£(LN,NAME)££(LN,NAME  )£(LN COMMA MISSING)£(LN,)'                                   487
'                                                                                   488
**************************************************************************        489
££(PS,----------------------------------------------------------------------)     490
£(PS,*******************TEST-3 LN FUNCTION*************************************)'  491
£(RS)£(RS)£(RS)£(RS)£(RS)£(RS)£(D£(PS,THE DEFINITION OF FORMS BEGINS)S,£(RS))       492
£(DS,,ABC IS MY ABDD) £(SS,,ABC,ABDD)          £(DS,£(RS))£(DS,£(RS))£(DS,£(RS     493
                              ))££(SS,X2,MARKER1,MARKER2,MARKER3)££(SS,X4,PARI      494
```

```
S,LEARNING,FRENCH)££(SS,X1,SMITH,SEAT,HOUSE,COMMONS)£(RS)£(RS)£(RS)£(RS)£(RS)£(R
S)£(SS,X3,MARKER1,MARKER2,MARKER3,MARKER4)£(RS)£(RS)£(PS,£(CL,))£(PS,£(CL,,))£(P
S,£(CL,,THIS,BOOK))'
£(PS,                    STORE  OF  FORMS)'
£(PS,**************************************************************************)'
£(PS,             X1  MR SMITH HAS A SEAT IN THE HOUSE OF COMMONS)'
£(PS,             X2  THE CLASSIC ROUTE TO THE MIDNIGHT SUN)'
£(PS,             X3  AFTER LUNCH YOU CONTINUE BY TRAIN A ROOM IS RESERVED FOR Y
OU)'
£(PS,             X4  HE WENT TO PARIS WITH THE INTENTION OF LEARNING FRENCH)'
X1,MR SMITH HAS A SEAT IN THE HOUSE OF COMMONS'
X2,MARKER1 CLASSIC MARKER2 TO MARKER1 MARKER3'
X3,AFTER LUNCH MARKER1 CONTINUE MARKER3 MARKER4 MARKER2 A ROOM IS RESERVED FOR M
ARKER1'
X4,HE WENT TO PARIS WITH THE INTENTION OF LEARNING FRENCH'
£(PS,£(CLX1,TAYLOR,FLAT,CITY,LONDON)£(CL,X1,TAYLOR,FLAT,CITY,LONDON))'
£(PS,£(CL,X2))'
£(PS,£(CL,X2,THE,ROUTE,MIDNIGHT SUN))'
£(PS,£(CL,X2,))'
£(PS,£(CL,X3,YOU,BY TRAIN))'
£(PS,£(CL,X5))'
£(PS,£(CL,X3,YOU,BY TRAIN))'
£(PS,£(CL,X4,STOCKHOLM,MEETING,FRIENDS,TELEIE,COSTA))'
£(LN,NAME)££(LN,NAME  )£(LN COMMA MISSING)£(LN,)'
'
**************************************************************************
££(PS,-------------------------------------------------------------------)
£(PS,*******************TEST-4 LN FUNCTION*********************************)'
£(DS,,££(RS))£(SS,,MARKER1,MARKER2,MARKER3,MARKER4,MARKER5),££(PS,£(CL,,DING,BIR
DS,LOVE)),££(DS,XZ,££(RS))£(SS,XZ,MARKER1,MARKER2)£(PS,£(CL,XZ,THOUGH YOU DO,SO
TRUE A FOOL IS))'
WHEN MARKER2 DO MARKER4 SING MARKER5 HEY MARKER1 A MARKER1 MARKER1(,)SWEET MARKE
R3RS MARKER3 THE MARKER5 SPRING MARKER4.'
MARKER2 LOVE(,)THAT IN YOUR WILL(,)MARKER1 ANYTHING(,)HE THINKS NO   ILL. W.SHAK
SPEARE'
£(LN,NAME)££(LN,NAME  )£(LN COMMA MISSING)£(LN,)'
£(PS,-------------------------------------------------------------------)'
'
```

```
**********************************************************************************  533
£(PS,----------------------------------------------------------------------------)  534
£(PS,**********TEST-5 LN FUNCTION**********************************)'                 535
£(DS,XYZ,£(RS))£(DS,XYW,££(RS))£(SS,XYZ,1,2,3)£(PS,££(CL,XYZ,SO LONG,CAN,THIS))£   536
(SS,XYW,4,5,6)£(PS,££(CL,XYW,FORGET NOT YET,TH,NT))'                               537
1 AS MEN 2 BREATHE OR EYES 2 SEE(,) 1 LIVES 3(,) AND 3 GIVES LIFE TO THEE.'        538
4 5E  TRIED I6E6  OF SUCH A  TRU5 AS I HAVE MEA6 , MY  GREAT TRAVAIL SO GLADLY S   539
PE6,4.'                                                                            540
£(LN,NAME)££(LN,NAME  )£(LN COMMA MISSING)£(LN,)'                                  541
£(PS,----------------------------------------------------------------------)'     542
'                                                                                  543
**********************************************************************************  544
£(PS,----------------------------------------------------------------------------)  545
£(PS,**********TEST-6 LN FUNCTION**********************************)'                 546
£(DS,A1,££(RS))££(SS,A1,POOR,HAST THOU GOLDEN SLUMBERS,SWEET CONTENT)£(PS,££(CL,   547
A1))£(DS,PROG,(£(PS,£(CL,£(RS)))))£(CL,PROG)£(LN,NAME)££(LN,NAME  )£(LN COMMA MI   548
SSING)£(LN,)'                                                                      549
ART THOU POOR(,)YET HAST THOU GOLDEN SLUMBERS(,)O SWEET CONTENT.'                  550
A1,RICH,IS THY MIND PERPLEXED,PUNISHMENT'                                          551
£(PS,----------------------------------------------------------------------)'     552
'                                                                                  553
**********************************************************************************  554
£(PS,----------------------------------------------------------------------------)  555
£(PS,**********TEST-7 LN FUNCTION**********************************)'                 556
£(DS,A,££(RS))£(SS,A,TAKE,DIE),£(DS,B,(£(CL,A,GIVE,LIVE))),£(PS,(£(CL,B))),£(PS,   557
££(CL,B))£(PS,£(CL,B))£(LN,NAME)££(LN,NAME  )£(LN COMMA MISSING)£(LN,)£(PS,-----   558
----------------------------------------------------------------------)'THE        559
THESE DELIGHTS IF THOU CANST TAKE(,)MIRTH(,)WITH THEE I MEAN TO DIE.'              560
£(PS,----------------------------------------------------------------------)'     561
'                                                                                  562
**********************************************************************************  563
£(PS,----------------------------------------------------------------------------)  564
£(PS,**********TEST-8 LN FUNCTION**********************************)'                 565
£(DS,SONG,£(RS))£(SS,SONG,SOME,THING,ELSE)£(DS,SING,(£(PS,£(CL,SONG,£(RS)))£(CL,   566
SING)))£(CL,SING)'                                                                 567
SOME THING WENT TO MOW WENT TO MOW A MEADOW ELSE AND HIS DOG WENT TO MOW A MEADO   568
W'                                                                                 569
ONE,MAN,ONE MAN'                                                                   570
```

```
TWO,MEN,(TWO MEN  ONE MAN)'                                                    571
THREE,MEN,(THREE MEN  TWO MEN  ONE MAN)'                                       572
£(LN,NAME)£(LN,NAME  )£(LN COMMA MISSING)£(LN,)'                               573
£(PS,----------)'                                                              574
'                                                                              575
                                                                               576
          GROUP-5  PROGRAMS FOR TESTING PF FUNCTION                            577
                                                                               578
**********                                                                     579
££(PS,----------)                                                              580
£(PS,**********TEST-1 PF FUNCTION**********)'                                  581
£(DS,TEXT,££(RS))£(SS,TEXT,THAT,THE,IN,COINCIDE)£(PF,NAME)£(PF,TEXT )£(PF,TEXT)£ 582
(PF,)£(DS,,£(RS))£(PF,)£(SS,,OF,ABNORMALITIES)£(PF,)£(DS,AB,£(RS)),£(PF,AB),£(SS 583
,AB,THIS,CHARACTERS)£(PF,AB)'                                                  584
IT HAS THEREFORE BEEN SUGGESTED THAT MARY WAS ACTUALLY BORN ON THE 7TH AND THAT  585
THE DATE WAS ALTERED TO THE 8TH IN ORDER TO COINCIDE WITH THE FEAST OF THE VIRGI 586
N.'                                                                            587
DURING THE WHOLE OF THIS PERIOD(,)HAMILTON BLOOD WAS GENERALLY CONSIDERED A CONV 588
ENIENT SCAPEGOAT ON WHICH TO BLAME ABNORMALITIES OF TEMPER.''                  589
THIS STRING HAS LESS THAN 80 CHARACTERS'                                       590
'                                                                              591
**********                                                                     592
££(PS,----------)                                                              593
£(PS,**********TEST-2 PF FUNCTION**********)'                                  594
£(DS,TEXT,££(RS))£(SS,TEXT,THAT,THE,IN,COINCIDE)£(PF,NAME)£(PF,TEXT )£(PF,TEXT)£ 595
(PF,)£(DS,,£(RS))£(PF,)£(SS,,OF,ABNORMALITIES)£(PF,)£(DS,AB,£(RS)),£(PF,AB),£(SS 596
,AB,THIS,CHARACTERS)£(PF,AB)£(PF AB)£(SS,TEXT,A,B,C,D,E,F,G,H,I,J,K,L,M,N)££(PF, 597
TEXT)£(SS,,A,B,C,D,E,F,G)££(PF,)'                                              598
IT HAS THEREFORE BEEN SUGGESTED THAT MARY WAS ACTUALLY BORN ON THE 7TH AND THAT  599
THE DATE WAS ALTERED TO THE 8TH IN ORDER TO COINCIDE WITH THE FEAST OF THE VIRGI 600
N.'                                                                            601
DURING THE WHOLE OF THIS PERIOD(,)HAMILTON BLOOD WAS GENERALLY CONSIDERED A CONV 602
ENIENT SCAPEGOAT ON WHICH TO BLAME ABNORMALITIES OF TEMPER.'                   603
THIS STRING HAS LESS THAN 80 CHARACTERS'                                       604
'                                                                              605
**********                                                                     606
££(PS,----------)                                                              607
£(PS,**********TEST-3 PF FUNCTION**********)'                                  608
```

```
£(DS,QQ,(£(PS,(SHE IS DESCRIBED AS VERY PRETTY INDEED))£(SS,QQ,££(RS))££(PF,QQ)
£(DS,QQ,££(CL,QQ,£(RS)))£(PF,QQ)£(CL,QQ)))£(CL,QQ)'
VERY PRETTY INDEED'
VERY CLEVER'
VERY CLEVER'
AWFULLY BORING'
'
**********************************************************************
££(PS,----------------------------------------------------------------)
£(PS,*******************TEST-4 PF FUNCTION*********************************)'
£(DS,,£(RS)) £(PF,)£(SS,,MARKER1,MARKER2,MARKER3,MARKER4,MARKER5,MARKER6,MARKER7
)££(PF,)£(PS,££(CL,,POSSESSED,CHILDHOOD,PASSION,MARY,OR FALSEHOOD))
£(DS,ABC,£(RS))£(PF,ABC)£(SS,ABC,MARKER1)££(PF,ABC)£(PS,££(CL,ABC,MARY QUEEN OF
SCOTS))'
BEING MARKER1 SINCE MARKER2 BY A MARKER3 FOR THE SUBJECT OF MARKER4 QUEEN OF SCO
TS(,) I WISHED TO TEST FOR MYSELF THE TRUTH MARKER5 OF THE MANY LEGENDS WHICH SU
RROUNDED HER NAME.'
FIND OUT WHAT MARKER1 MUST HAVE BEEN LIKE A PERSON.'
'
**********************************************************************
££(PS,----------------------------------------------------------------)
£(PS,*******************TEST-5 PF FUNCTION*********************************)'
£(DS,A1,£(RS))£(PF,A1)
£(DS,A2,£(RS))£(PF,A2)
£(SS,A1,SE STARK COND,BLEAK PARALLEL,AL CLIMATE WH)£(PF,A1)
£(SS,A2,FORMAL DANCE,THESE,THE PEOPLE,OF)£(PF,A2)'
THESE STARK CONDITIONS FOUND A BLEAK PARALLEL IN THE POLITICAL CLIMATE WHICH
THEN PREVAILED BETWEEN COUNTRIES.'
THESE ARRANGEMENTS(,) LIKE THE STEPS OF A FORMAL DANCE(,) CONVEY LITTLE OF THE F
EELINGS OF THE PEOPLE CONCERNED.'
££(CL,A1,SE STARK COND,BLEAK PARALLEL,AL CLIMATE WH)'
££(CL,A2,FORMAL DANCE,THESE,THE PEOPLE,OF)'
'
**********************************************************************
**********************************************************************
££(PS,----------------------------------------------------------------)
£(PS,*******************TEST-6 PF FUNCTION*********************************)'
£(DS,X1,£(RS))£(PF,X1)
```

```
£(SS,X1,SHE,HAD BEEN MARRIED,NINETEEN)£(PF,X1)                                  647
£(DS,PROGRAM,(£(PS,£(CL,£(RS)))))£(PF,PROGRAM)                                  648
£(CL,PROGRAM)                                                                   649
£(CL,PROGRAM)'                                                                  650
SHE HAD BEEN MARRIED AT THE AGE OF NINETEEN.'                                   651
X1,HE,DIED,55'                                                                  652
X1,HE,WAS MARRIED,76'                                                           653
'                                                                               654
**********************************************************************          655
££(PS,-------------------------------------------------------------)            656
£(PS,********************TEST-7 PF FUNCTION*************************************)'  657
£(DS,A,£(RS))£(PF,A)                                                            658
£(SS,A,JAMES AND MARY,HAPPY)£(PF,A)                                             659
£(DS,B,(£(CL,A,JOHN AND MARGARET,UNHAPPY)))£(PF,B)                              660
£(PS,(£(CL,B)))                                                                 661
£(PS,££(CL,B))                                                                  662
£(PS,£(CL,B)'                                                                   663
THE MARRIAGE OF JAMES AND MARY DOES NOT SEEM TO HAVE BEEN A PARTICULARLY HAPPY O  664
NE IN ITS EARLY STAGES.'                                                        665
'                                                                               666
**********************************************************************          667
££(PS,-------------------------------------------------------------)            668
£(PS,********************TEST-8 PF FUNCTION*************************************)'  669
£(PS,           **FORMS  IN  STORE**)'                                          670
£(PS,**********************************************************************)'  671
£(D£(PS,**THE DEFINITION OF FORMS BEGINS**)S,£(RS))£(PF,X1)£(DS,£(RS))£(PF,X2)£(  672
DS,£(RS))£(PF,X3)£(DS,£(RS))£(PF,X4)£(PF,X5)£(DS,£(RS))£(PF,)£(DS,£(RS))£(PF,X5)  673
£(SS,X1, W,TH,TAKEN,ON,NO)£(PF,X1)£(SS,X2,CH,A,L,M,NEY,I)£(PF,X2)£(SS,X3,SERVANT  674
S,PAYING,DEPENDED)£(PF,X3)£(SS,X4,OR,HE,NECESSITIES,FOU)£(PF,X4)£(SS,,A,B,C,D,E,  675
F)£(PF,)£(SS,X5,SEIZED,BUT,LATER)£(PF,X5)£(PS,££(CL,X1, W,TH,TAKEN,ON,NO)££(CL,X  676
2,CH,A,L,M,NEY,I)££(CL,X3,SERVANTS,PAYING,DEPENDED)££(CL,X4,OR,HE,NECESSITIES,FO  677
U)££(CL,,A,B,C,D,E,F)££(CL,X5,SEIZED,BUT,LATER))'                               678
X1,THE ONLY THING WHICH WAS NOT TAKEN FROM '                                    679
X2,MARY WAS HER ACTUAL MONEY(,) ON WHICH'                                       680
X3, SHE DEPENDED FOR PAYING HER SERVANTS AND '                                  681
X4,FOR HER OWN NECESSITIES.THIS SHE FOUND '                                     682
,STILL IN THE CUPBOARD WHERE SHE HAD LEFT IT '                                  683
X5,BUT LATER DIRECTIVES CAME FROM LONDON THAT THIS TOO WAS TO BE SEIZED.'        684
```

```
'                                                                                685
*********************************************************************************** 686
                                                                                    687
            GROUP-6  PROGRAMS FOR TESTING EQ FUNCTION                               688
                                                                                    689
*********************************************************************************** 690
£(PS,-------------------------------------------------------------------------)     691
£(PS,*******************TEST-1 EQ FUNCTION*************************************)     692
£(DS,£(RS),£(RS))£(DS,£(RS),£(RS))£(DS,EQUAL,(£(EQ,£(RS),LINE1,(£(PS,££(CL,LINE1  693
)))),(£(PS,££(CL,LINE2))))£(CL,EQUAL)))£(LN,NAME  )£(CL,EQUAL)'                    694
LINE1'                                                                              695
SPRING(,)THE SWEET SPRING(,)IS THE YEAR   S PLEASANT KING'                          696
LINE2'                                                                              697
THEN BLOOMS EACH THING(,)THEN MAIDS DANCE IN A RING.'                               698
LINE1'                                                                              699
LINE2'                                                                              700
ANY NAME'                                                                           701
LINE1'                                                                              702
£(PF,LINE1)£(PF,LINE2)£(PF,EQUAL)'                                                  703
'                                                                                   704
*********************************************************************************** 705
£(PS,-------------------------------------------------------------------------)     706
£(PS,*******************TEST-1 EQ FUNCTION**WITH TRACE*************************)     707
£(TN)'                                                                              708
£(DS,£(RS),£(RS))£(DS,£(RS),£(RS))£(DS,EQUAL,(£(EQ,£(RS),LINE1,(£(PS,££(CL,LINE1  709
)))),(£(PS,££(CL,LINE2))))£(CL,EQUAL)))£(LN,NAME  )£(CL,EQUAL)'                    710
LINE1'                                                                              711
SPRING(,)THE SWEET SPRING(,)IS THE YEAR   S PLEASANT KING'                          712
LINE2'                                                                              713
THEN BLOOMS EACH THING(,)THEN MAIDS DANCE IN A RING.'                               714
LINE1'                                                                              715
LINE2'                                                                              716
ANY NAME'                                                                           717
LINE1'                                                                              718
£(PF,LINE1)£(PF,LINE2)£(PF,EQUAL)'                                                  719
'                                                                                   720
*********************************************************************************** 721
£(PS,-------------------------------------------------------------------------)     722
```

```
£(PS,********************TEST-2 EQ FUNCTION************************************)  723
£(DS,TEXT,££(RS))£(DS,TEMP,££(CL,TEXT))£(SS,TEXT,£(RS))£(DS,PROC,(£(DS,TEXT,££(C  724
L,TEXT,£(RS)))£(EQ,££(CL,TEMP),££(CL,TEXT),(£(PS,END  OF PROGRAM)£(PS,TEMP******  725
****££(CL,TEMP))£(PS,TEXT**********££(CL,TEXT))),(£(PS,TEXT**********££(CL,TEXT)  726
)£(SS,TEXT,£(RS))£(CL,PROC)))))£(CL,PROC)'                                         727
NOT ALL THAT TEMPTS YOUR WANDERING EYES'                                           728
TEMPTS,WANDERING'                                                                  729
1,2'                                                                               730
1,2'                                                                               731
MM1,MM2'                                                                           732
MM1,MM2'                                                                           733
MARKER1,MARKER2'                                                                   734
MARKER1,MARKER2'                                                                   735
SLEEPING,PLEASING'                                                                 736
SLEEPING,PLEASING'                                                                 737
TEMPTS,WANDERING'                                                                  738
£(PS,-------------------------------------------------------------------------)'  739
'                                                                                  740
**********************************************************************************  741
£(PS,-------------------------------------------------------------------------)  742
£(PS,********************TEST-2 EQ FUNCTION**WITH  TRACE************************)  743
£(TN)'                                                                             744
£(DS,TEXT,££(RS))£(DS,TEMP,££(CL,TEXT))£(SS,TEXT,£(RS))£(DS,PROC,(£(DS,TEXT,££(C  745
L,TEXT,£(RS)))£(EQ,££(CL,TEMP),££(CL,TEXT),(£(PS,END  OF PROGRAM)£(PS,TEMP******  746
****££(CL,TEMP))£(PS,TEXT**********££(CL,TEXT))),(£(PS,TEXT**********££(CL,TEXT)  747
)£(SS,TEXT,£(RS))£(CL,PROC)))))£(CL,PROC)'                                         748
NOT ALL THAT TEMPTS YOUR WANDERING EYES'                                           749
TEMPTS,WANDERING'                                                                  750
1,2'                                                                               751
1,2'                                                                               752
MM1,MM2'                                                                           753
MM1,MM2'                                                                           754
MARKER1,MARKER2'                                                                   755
MARKER1,MARKER2'                                                                   756
SLEEPING,PLEASING'                                                                 757
SLEEPING,PLEASING'                                                                 758
TEMPTS,WANDERING'                                                                  759
£(PS,-------------------------------------------------------------------------)'  760
```

```
'                                                                                761
*********************************************************************************** 762
 (PS,-------------------------------------------------------------------------)    763
£(PS,*******************TEST-3 EQ FUNCTION*************************************)    764
£(DS,V1,££(RS))£(DS,V2,££(RS))£(DS,V3,££(RS))££(SS,V3,1,2,3)£(DS,EQUAL,(£(EQ,£(S    765
S,V1,£(RS))££(CL,V1,£(RS)),££(CL,V2),(£(PS,££(CL,V3,£(RS)))£(PS,££(CL,V2))),(£(D    766
S,V1,££(CL,V1,£(RS)))£(PS,££(CL,V1))£(CL,EQUAL)))))£(CL,EQUAL)'                    767
AND II ILKA BUSH JJ AROUND LL HINGING'                V1                         768
AND SEE THE BIRDS ON ILKA BUSH AND LEAVES AROUND THEM HINGING'  V2               769
DOWN 1 SWEET 2 BONNIE 3 LILY'1=IN YON GARDEN,2=AND GAY WHERE,3=GROWS THE         770
II,JJ,LL'SEGMENT V1                                                              771
I1,J,L1'    ARGUMENTS IN CALL FUNCTION ,V1 IS THE CALLED FORM.                   772
I1,J,L1' REDEFINE V1 MARKERS SUBSTITUTED WITH THE ARG. OF CALL FUNCTION          773
I1,J,L1'                                                                         774
II,JJ,LL'                                                                        775
II,JJ,LL'                                                                        776
II,JJ,LL'                                                                        777
III,JJJ,LLL'                                                                     778
III,JJJ,LLL'                                                                     779
III,JJJ,LLL'                                                                     780
IIII,JJJJ,LLLL'                                                                  781
IIII,JJJJ,LLLL'                                                                  782
IIII,JJJJ,LLLL'                                                                  783
SEE THE BIRDS ON,AND LEAVES,THEM'                                                784
IN YON GARDEN,AND GAY WHERE,GROWS THE'                                           785
£(PS,-------------------------------------------------------------------------)' 786
'                                                                                787
*********************************************************************************** 788
£(PS,-------------------------------------------------------------------------)    789
£(PS,*******************TEST-3 EQ FUNCTION***WITH  TRACE************************)    790
£(TN)'                                                                           791
£(DS,V1,££(RS))£(DS,V2,££(RS))£(DS,V3,££(RS))££(SS,V3,1,2,3)£(DS,EQUAL,(£(EQ,£(S    792
S,V1,£(RS))££(CL,V1,£(RS)),££(CL,V2),(£(PS,££(CL,V3,£(RS)))£(PS,££(CL,V2))),(£(D    793
S,V1,££(CL,V1,£(RS)))£(PS,££(CL,V1))£(CL,EQUAL)))))£(CL,EQUAL)'                    794
AND II ILKA BUSH JJ AROUND LL HINGING'                V1                         795
AND SEE THE BIRDS ON ILKA BUSH AND LEAVES AROUND THEM HINGING'  V2               796
DOWN 1 SWEET 2 BONNIE 3 LILY'1=IN YON GARDEN,2=AND GAY WHERE,3=GROWS THE         797
II,JJ,LL'SEGMENT V1                                                              798
```

```
I1,J,L1'    ARGUMENTS IN CALL FUNCTION ,V1 IS THE CALLED FORM.                  799
I1,J,L1' REDEFINE V1 MARKERS SUBSTITUTED WITH THE ARG. OF CALL FUNCTION         800
I1,J,L1'                                                                        801
II,JJ,LL'                                                                       802
II,JJ,LL'                                                                       803
II,JJ,LL'                                                                       804
III,JJJ,LLL'                                                                    805
III,JJJ,LLL'                                                                    806
III,JJJ,LLL'                                                                    807
IIII,JJJJ,LLLL'                                                                 808
IIII,JJJJ,LLLL'                                                                 809
IIII,JJJJ,LLLL'                                                                 810
SEE THE BIRDS ON,AND LEAVES,THEM'                                               811
IN YON GARDEN,AND GAY WHERE,GROWS THE'                                          812
£(PS,----------)'                                                               813
'                                                                               814
**********                                                                      815
£(PS,----------)                                                                816
£(PS,**********TEST-4 EQ FUNCTION**********)                                    817
£(DS,COUNTER,IIIII)£(DS,UNIT,I)£(DS,QQQQ,(£(PS,WHEN MU1 MA2 YOU3 IN EARLY GREECE 818
 SHE SUNG)£(PS,QQQQ*****££(CL,QQQQ))£(SS,QQQQ,£(RS))£(DS,QQQQ,££(CL,QQQQ,£(RS))) 819
£(EQ,££(CL,COUNTER),££(CL,UNIT),(£(PS,END PROGRAM)£(PS,QQQQ*******££(CL,QQQQ))£( 820
PS,COUNTER****££(CL,COUNTER))£(PS,UNIT*******££(CL,UNIT))),(£(DS,UNIT,I££(CL,UNI 821
T))£(PS,UNIT*****££(CL,UNIT))£(CL,QQQQ)))))£(CL,QQQQ)'                          822
1,2,3'SUBSTITUTED BY MARKERS                                                    823
M1,M2,M3'ARGUMENTS IN CALL FUNCTION,SUBSTITUTE MARKERS                          824
M1,M2,M3'SUBSTITUTED BY MARKERS                                                 825
MM1,MM2,MM3'                                                                    826
MM1,MM2,MM3'                                                                    827
MMM1,MMM2,MMM3'                                                                 828
MMM1,MMM2,MMM3'                                                                 829
MMMM1,MMMM2,MMMM3'                                                              830
MMMM1,MMMM2,MMMM3'                                                              831
SIC(,)HEAVENLY,ID(,)WAS,NG(,)WHILE YET'                                         832
£(PS,QQQQ**********£(CL,QQQQ))'                                                 833
£(PS,----------)'                                                               834
'                                                                               835
**********                                                                      836
```

```
£(PS,----------)                                                                  837
£(PS,**********TEST-4 EQ FUNCTION**WITH  TRACE**********)                        838
£(TN)'                                                                            839
£(DS,COUNTER,IIIII)£(DS,UNIT,I)£(DS,QQQQ,(£(PS,WHEN MU1 MA2 YOU3 IN EARLY GREECE  840
 SHE SUNG)£(PS,QQQQ*****££(CL,QQQQ))£(SS,QQQQ,£(RS))£(DS,QQQQ,££(CL,QQQQ,£(RS)))  841
£(EQ,££(CL,COUNTER),££(CL,UNIT),(£(PS,END PROGRAM)£(PS,QQQQ*******££(CL,QQQQ))£(  842
PS,COUNTER****££(CL,COUNTER))£(PS,UNIT*******££(CL,UNIT))),(£(DS,UNIT,I££(CL,UNI  843
T))£(PS,UNIT*****££(CL,UNIT))£(CL,QQQQ)))))£(CL,QQQQ)'                           844
1,2,3'SUBSTITUTED BY MARKERS                                                      845
M1,M2,M3'ARGUMENTS IN CALL FUNCTION,SUBSTITUTE MARKERS                            846
M1,M2,M3'SUBSTITUTED BY MARKERS                                                   847
MM1,MM2,MM3'                                                                      848
MM1,MM2,MM3'                                                                      849
MMM1,MMM2,MMM3'                                                                   850
MMM1,MMM2,MMM3'                                                                   851
MMMM1,MMMM2,MMMM3'                                                                852
MMMM1,MMMM2,MMMM3'                                                                853
SIC(,)HEAVENLY,ID(,)WAS,NG(,)WHILE YET'                                           854
£(PS,QQQQ**********£(CL,QQQQ))'                                                   855
£(PS,----------)'                                                                 856
'                                                                                 857
**********                                                                        858
£(PS,----------)                                                                  859
£(PS,**********TEST-5 EQ FUNCTION**********)'                                     860
£(DS,XYZ,£(RS))£(SS,XYZ,£(RS))£(DS,END-PROC,£(RS))£(DS,£(RS),£(RS))£(SS,£(RS),£(  861
RS))£(DS,OTHER,(£(EQ,££(CL,ZYX,£(RS)),££(CL,END-PROC),(£(PS,END OF PROGRAM)£(LN,  862
NAME**********)),(£(PS,££(CL,ZYX,£(RS)))))£(CL,OTHER)))£(DS,ALTER,(£(EQ,££(CL,XY  863
Z,£(RS)),££(CL,END-PROC),(£(PS,XYZ*******££(CL,XYZ,£(RS)))£(PS,END-PROC**££(CL,E  864
ND-PROC))£(CL,OTHER)),(£(PS,££(CL,XYZ,£(RS)))£(CL,ALTER)))))£(CL,ALTER)'          865
THIS IS A CAT'                                                                    866
CAT'                                                                              867
THIS IS A HORSE'                                                                  868
ZYX'                                                                              869
THIS IS A CAT'                                                                    870
ZYX'                                                                              871
CAT'                                                                              872
CAT'                                                                              873
CAT'                                                                              874
```

```
DOG'                                                                                                                    875
DOG'                                                                                                                    876
MOUSE'                                                                                                                  877
MOUSE'                                                                                                                  878
SHEEP'                                                                                                                  879
SHEEP'                                                                                                                  880
COW'                                                                                                                    881
COW'                                                                                                                    882
HORSE'                                                                                                                  883
HORSE'                                                                                                                  884
CAT'                                                                                                                    885
CAT'                                                                                                                    886
DOG'                                                                                                                    887
DOG'                                                                                                                    888
MOUSE'                                                                                                                  889
MOUSE'                                                                                                                  890
SHEEP'                                                                                                                  891
SHEEP'                                                                                                                  892
COW'                                                                                                                    893
COW'                                                                                                                    894
HORSE'                                                                                                                  895
'                                                                                                                       896
**********************************************************************************************************************  897
£(PS,----------------------------------------------------------------------------------------------------)              898
£(PS,********************TEST-5 EQ FUNCTION**WITH TRACE*************************)                                      899
£(TN)'                                                                                                                  900
£(DS,XYZ,£(RS))£(SS,XYZ,£(RS))£(DS,END-PROC,£(RS))£(DS,£(RS),£(RS))£(SS,£(RS),£(                                        901
RS))£(DS,OTHER,(£(EQ,££(CL,ZYX,£(RS)),££(CL,END-PROC),(£(PS,END OF PROGRAM)£(LN,                                        902
NAME**********)),(£(PS,££(CL,ZYX,£(RS)))))£(CL,OTHER)))£(DS,ALTER,(£(EQ,££(CL,XY                                       903
Z,£(RS)),££(CL,END-PROC),(£(PS,XYZ*******££(CL,XYZ,£(RS)))£(PS,END-PROC**££(CL,E                                        904
ND-PROC))£(CL,OTHER)),(£(PS,££(CL,XYZ,£(RS)))£(CL,ALTER))))£(CL,ALTER)'                                                 905
THIS IS A CAT'                                                                                                          906
CAT'                                                                                                                    907
THIS IS A HORSE'                                                                                                        908
ZYX'                                                                                                                    909
THIS IS A CAT'                                                                                                          910
ZYX'                                                                                                                    911
CAT'                                                                                                                    912
```

```
CAT'                                                                                                    913
CAT'                                                                                                    914
DOG'                                                                                                    915
DOG'                                                                                                    916
MOUSE'                                                                                                  917
MOUSE'                                                                                                  918
SHEEP'                                                                                                  919
SHEEP'                                                                                                  920
COW'                                                                                                    921
COW'                                                                                                    922
HORSE'                                                                                                  923
HORSE'                                                                                                  924
CAT'                                                                                                    925
CAT'                                                                                                    926
DOG'                                                                                                    927
DOG'                                                                                                    928
MOUSE'                                                                                                  929
MOUSE'                                                                                                  930
SHEEP'                                                                                                  931
SHEEP'                                                                                                  932
COW'                                                                                                    933
COW'                                                                                                    934
HORSE'                                                                                                  935
'                                                                                                       936
**********************************************************************************************        937
£(PS,----------------------------------------------------------------------------------)               938
£(PS,********************TEST-6 EQ FUNCTION*************************************)                     939
£(DS,EXPRESSION,AEXPONENT2 SIGN1ZTIMESATIMESB SIGN2BEXPONENT2)'                                         940
£(DS,TEST,A**2 -2*A*B +B**2)'                                                                           941
£(SS,EXPRESSION,EXPONENT,TIMES,SIGN1,SIGN2)'                                                            942
£(EQ,££(CL,TEST),££(CL,EXPRESSION,**,*,£(RS),£(RS)),(£(PS,END PROGRAM)£(PS,EXPRE                        943
SSION*=££(CL,EXPRESSION,**,*,£(RS),£(RS)))£(PS,TEST*******=££(CL,TEST))),(£(PS,E                        944
XPRESSION*=££(CL,EXPRESSION,**,*,£(RS),£(RS))£(PS,TEST*******=££(CL,TEST))))'                           945
+'                                                                                                      946
+'                                                                                                      947
+'                                                                                                      948
+'                                                                                                      949
£(EQ,££(CL,TEST),££(CL,EXPRESSION,**,*,£(RS),£(RS)),(£(PS,END PROGRAM)£(PS,EXPRE                        950
```

```
SSION*=££(CL,EXPRESSION,**,*,£(RS),£(RS)))£(PS,TEST*******=££(CL,TEST))),(£(PS,E  951
XPRESSION*=££(CL,EXPRESSION,**,*,£(RS),£(RS))£(PS,TEST*******=££(CL,TEST))))'    952
+'                                                                                953
 '                                                                                954
+'                                                                                955
-'                                                                                956
£(EQ,££(CL,TEST),££(CL,EXPRESSION,**,*,£(RS),£(RS)),(£(PS,END PROGRAM)£(PS,EXPRE  957
SSION*=££(CL,EXPRESSION,**,*,£(RS),£(RS)))£(PS,TEST*******=££(CL,TEST))),(£(PS,E  958
XPRESSION*=££(CL,EXPRESSION,**,*,£(RS),£(RS))£(PS,TEST*******=££(CL,TEST))))'    959
-'                                                                                960
-'                                                                                961
-'                                                                                962
-'                                                                                963
£(EQ,££(CL,TEST),££(CL,EXPRESSION,**,*,£(RS),£(RS)),(£(PS,END PROGRAM)£(PS,EXPRE  964
SSION*=££(CL,EXPRESSION,**,*,£(RS),£(RS)))£(PS,TEST*******=££(CL,TEST))),(£(PS,E  965
XPRESSION*=££(CL,EXPRESSION,**,*,£(RS),£(RS))£(PS,TEST*******=££(CL,TEST))))'    966
-'                                                                                967
+'                                                                                968
-'                                                                                969
+'                                                                                970
'                                                                                971
*********************************************************************************  972
£(PS,-------------------------------------------------------------------------)  973
£(PS,*******************TEST-6 EQ FUNCTION**WITH TRACE************************)  974
£(TN)'                                                                           975
£(DS,EXPRESSION,AEXPONENT2 SIGN12TIMESATIMESB SIGN2BEXPONENT2)'                  976
£(DS,TEST,A**2 -2*A*B +B**2)'                                                    977
£(SS,EXPRESSION,EXPONENT,TIMES,SIGN1,SIGN2)'                                     978
£(EQ,££(CL,TEST),££(CL,EXPRESSION,**,*,£(RS),£(RS)),(£(PS,END PROGRAM)£(PS,EXPRE  979
SSION*=££(CL,EXPRESSION,**,*,£(RS),£(RS)))£(PS,TEST*******=££(CL,TEST))),(£(PS,E  980
XPRESSION*=££(CL,EXPRESSION,**,*,£(RS),£(RS))£(PS,TEST*******=££(CL,TEST))))'    981
+'                                                                                982
+'                                                                                983
+'                                                                                984
+'                                                                                985
£(EQ,££(CL,TEST),££(CL,EXPRESSION,**,*,£(RS),£(RS)),(£(PS,END PROGRAM)£(PS,EXPRE  986
SSION*=££(CL,EXPRESSION,**,*,£(RS),£(RS)))£(PS,TEST*******=££(CL,TEST))),(£(PS,E  987
XPRESSION*=££(CL,EXPRESSION,**,*,£(RS),£(RS))£(PS,TEST*******=££(CL,TEST))))'    988
```

```
+'                                                                                                    989
 '                                                                                                    990
+'                                                                                                    991
-'                                                                                                    992
£(EQ,££(CL,TEST),££(CL,EXPRESSION,**,*,£(RS),£(RS)),(£(PS,END PROGRAM)£(PS,EXPRE                      993
SSION*=££(CL,EXPRESSION,**,*,£(RS),£(RS)))£(PS,TEST*******=££(CL,TEST))),(£(PS,E                      994
XPRESSION*=££(CL,EXPRESSION,**,*,£(RS),£(RS))£(PS,TEST*******=££(CL,TEST))))'                         995
-'                                                                                                    996
-'                                                                                                    997
-'                                                                                                    998
-'                                                                                                    999
£(EQ,££(CL,TEST),££(CL,EXPRESSION,**,*,£(RS),£(RS)),(£(PS,END PROGRAM)£(PS,EXPRE                      1000
SSION*=££(CL,EXPRESSION,**,*,£(RS),£(RS)))£(PS,TEST*******=££(CL,TEST))),(£(PS,E                      1001
XPRESSION*=££(CL,EXPRESSION,**,*,£(RS),£(RS))£(PS,TEST*******=££(CL,TEST))))'                         1002
-'                                                                                                    1003
+'                                                                                                    1004
-'                                                                                                    1005
+'                                                                                                    1006
'                                                                                                    1007
**************************************************************************************************  1008
£(PS,----------------------------------------------------------------------------)                   1009
£(PS,********************TEST-7 EQ FUNCTION*************************************)                   1010
£(DS,P1,(£(PS,£(RS))))'                                                                              1011
£(DS,END,(£(LN,NAME  )£(PS,END OF PROGRAM)))'                                                        1012
£(DS,X,(EQ,£(RS),END-PR,))'                                                                          1013
£(DS,Y,((£(CL,END)),(£(CL,P1))))'                                                                    1014
£(DS,A,(£(£(CL,X)£(CL,Y))£(CL,A)))'                                                                  1015
£(CL,A)'                                                                                             1016
FIRST EXECUTION'                                                                                     1017
A WISE MAN WILL MAKE MORE OPPORTUNITIES THAN HE FINDS'                                               1018
SECOND EXECUTION'                                                                                    1019
MONEY IS LIKE MUCK (,) NOT GOOD EXCEPT IT CAN BE SPREAD'                                             1020
THIRD EXECUTION'                                                                                     1021
THE REMEDY IS WORSE THAN THE DISEASE'                                                                1022
FOURTH EXECUTION'                                                                                    1023
STUDIES SERVE FOR DELIGHT(,)FOR ORNAMENT(,)AND FOR ABILITY'                                          1024
END-PR'                                                                                              1025
'                                                                                                    1026
```

```
********************************************************************************* 1027
£(PS,-------------------------------------------------------------------------) 1028
£(PS,********************TEST-7 EQ FUNCTION**WITH TRACE*************************) 1029
£(TN)' 1030
£(DS,P1,(£(PS,£(RS))))' 1031
£(DS,END,(£(LN,NAME  )£(PS,END OF PROGRAM)))' 1032
£(DS,X,(EQ,£(RS),END-PR,))' 1033
£(DS,Y,((£(CL,END)),(£(CL,P1))))' 1034
£(DS,A,(£(£(CL,X)£(CL,Y))£(CL,A)))' 1035
£(CL,A)' 1036
FIRST EXECUTION' 1037
A WISE MAN WILL MAKE MORE OPPORTUNITIES THAN HE FINDS' 1038
SECOND EXECUTION' 1039
MONEY IS LIKE MUCK (,) NOT GOOD EXCEPT IT CAN BE SPREAD' 1040
THIRD EXECUTION' 1041
THE REMEDY IS WORSE THAN THE DISEASE' 1042
FOURTH EXECUTION' 1043
STUDIES SERVE FOR DELIGHT(,)FOR ORNAMENT(,)AND FOR ABILITY' 1044
END-PR' 1045
' 1046
********************************************************************************* 1047
£(PS,-------------------------------------------------------------------------) 1048
£(PS,********************TEST-8 EQ FUNCTION*************************************) 1049
£(DS,Q,(£(DS,S,X)£(SS,S,Y)£(DS,Z,££(CL,S,A))£(EQ,££(CL,S,Y),££(CL,Z),££(CL,Z),(£ 1050
(CL,Q,££(CL,Z),Y)))))£(SS,Q,X,Y)' 1051
£(PS,£(CL,Q,ABCBCC,ABC))' 1052
£(PS,-------------------------------------------------------------------------)' 1053
' 1054
********************************************************************************* 1055
£(PS,-------------------------------------------------------------------------) 1056
£(PS,********************TEST-8 EQ FUNCTION**WITH  TRACE************************) 1057
£(TN)' 1058
£(DS,Q,(£(DS,S,X)£(SS,S,Y)£(DS,Z,££(CL,S,A))£(EQ,££(CL,S,Y),££(CL,Z),££(CL,Z),(£ 1059
(CL,Q,££(CL,Z),Y)))))£(SS,Q,X,Y)' 1060
£(LN,NAME  )£(PF,Q)£(PF,S)£(PF,Z)' 1061
£(PS,£(CL,Q,ABCBCC,ABC))' 1062
£(PS,-------------------------------------------------------------------------)' 1063
' 1064
```

IIC.10 Execution of TRAC Applications

. . . pages 224-309

```
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++                                    1
--------------------------------------------------------------------------------  2
**************TEST-1 FUNCTIONS  RS*PS*CL*DS*SS*************************          3
£(CL,A2)                                                                         4
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                5
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                  6
£(CL,A1,MOUNTAINS,FOUNTAINS,GRAVES)                                              7
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                8
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                  9
OVER THE MOUNTAINS AND OVER THE WAVES,UNDER THE FOUNTAINS AND UNDER THE GRAVES   10
--------------------------------------------------------------------------------  11
+++++THIS PS FUNCTION HAS NULL STRING+++++                                        12
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                     13
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                     14
--------------------------------------------------------------------------------  15
**************TEST-2 FUNCTIONS  RS*PS*CL*DS*SS*************************          16
+++++THIS PS FUNCTION HAS NULL STRING+++++                                        17
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                     18
UNDER FLOODS THAT ARE DEEPEST,WHICH NEPTUNE OBEY                                 19
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                     20
OVER ROCKS THAT ARE STEEPEST LOVE WILL FIND OUT THE WAY                          21
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                     22
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                23
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                   24
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                   25
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                26
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                   27
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                   28
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                   29
UNDER  THAT ARE DEEPEST,WHICH  OBEY*****OVER  THAT ARE  LOVE WILL FIND OUT THE  30
--------------------------------------------------------------------------------  31
+++++THIS PS FUNCTION HAS NULL STRING+++++                                        32
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                     33
                                                                                 34
                                                                                 35
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                     36
--------------------------------------------------------------------------------  37
**************TEST-3 FUNCTIONS  RS*PS*CL*DS*SS*************************          38
```

```
+++++THIS PS FUNCTION HAS NULL STRING+++++                                39
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                             40
£(CL,PCEM)                                                               41
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                             42
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                         43
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++          44
£(RS)                                                                    45
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                             46
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                         47
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++          48
WHEN I THE DAWN USED TO ADMIRE(,)AND PRAISED THE COMING DAY              49
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                             50
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                         51
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++          52
+++++THIS PS FUNCTION HAS NULL STRING+++++                                53
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                             54
+++++THIS PS FUNCTION HAS NULL STRING+++++                                55
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                             56
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                         57
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++           58
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++           59
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++           60
WHEN I THE  USED TO (,)AND  THE COMING DAY                               61
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                             62
WHEN I THE 1 USED TO 3,AND 2 THE COMING DAY                              63
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                             64
WHEN I THE MARKER1 USED TO MARKER3,AND MARKER2 THE COMING DAY            65
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                             66
WHEN I THE DAWN USED TO ADMIRE,AND PRAISED THE COMING DAY                67
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                             68
-------------------------------------------------------------------------69
+++++THIS PS FUNCTION HAS NULL STRING+++++                                70
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                             71
                                                                         72
                                                                         73
                                                                         74
                                                                         75
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                             76
```

```
------------------------------------------------------------------------ 77
**************TEST-4 FUNCTIONS  RS*PS*CL*DS*SS************************** 78
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                         79
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++           80
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++           81
TEXT**I LITTLE  THE GROWING FIRE  MY REST AWAY                           82
TEXT WITH MARKERS**I LITTLE 1 THE GROWING FIRE 2 MY REST AWAY            83
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                         84
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++          85
NEW-TEXT**I LITTLE THOUGHT THE GROWING FIRE MUST TAKE MY REST AWAY        86
+++++THIS PS FUNCTION HAS NULL STRING+++++                               87
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                            88
------------------------------------------------------------------------ 89
+++++THIS PS FUNCTION HAS NULL STRING+++++                               90
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                            91
                                                                         92
                                                                         93
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                            94
------------------------------------------------------------------------ 95
**************TEST-5 FUNCTIONS  RS*PS*CL*DS*SS************************** 96
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                         97
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++          98
YOUR     CHARMS IN HARMLESS       CHILDHOOD  LAY LIKE METALS IN THE     MINE***  99
YOUR CHARMS IN HARMLESS CHILDHOOD LAY LIKE METALS IN THE MINE           100
AGE FROM NO FACE TOOK MORE AWAY                                         101
+++++THIS PS FUNCTION HAS NULL STRING+++++                              102
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                           103
------------------------------------------------------------------------ 104
+++++THIS PS FUNCTION HAS NULL STRING+++++                              105
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                           106
                                                                        107
                                                                        108
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                           109
------------------------------------------------------------------------ 110
**************TEST-6 FUNCTIONS  RS*PS*CL*DS*SS************************** 111
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                        112
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++         113
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                        114
```

```
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                  115
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                116
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                  117
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                118
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                  119
£(DS,X,YOU COMMON PEOPLE OF THE SKIES(,)WHAT ARE YOU(,)WHEN THE MOON SHALL RISE  120
)£(DS,X,YOU COMMON PEOPLE OF THE SKIES(,)WHAT ARE YOU(,)WHEN THE MOON SHALL RIS  121
E)                                                                               122
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                123
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                  124
X****YOU COMMON PEOPLE OF THE SKIES,WHAT ARE YOU,WHEN THE MOON SHALL RISE        125
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                126
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                  127
N*****(£(DS,X,YOU COMMON PEOPLE OF THE SKIES(,)WHAT ARE YOU(,)WHEN THE MOON SHA  128
LL RISE))£(DS,X,YOU COMMON PEOPLE OF THE SKIES(,)WHAT ARE YOU(,)WHEN THE MOON S  129
HALL RISE)(£(DS,X,YOU COMMON PEOPLE OF THE SKIES(,)WHAT ARE YOU(,)WHEN THE MOON  130
 SHALL RISE))                                                                    131
---------------------------------------------------------------------------------132
+++++THIS PS FUNCTION HAS NULL STRING+++++                                        133
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                    134
                                                                                 135
                                                                                 136
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                    137
---------------------------------------------------------------------------------138
**************TEST-7 FUNCTIONS  RS*PS*CL*ES*SS**************************        139
XY*****SO WELL YOUR WORDS HIS NOBLE VIRTUES PRAISE,THAT ALL BOTH JUDGE YOU TO    140
RELATE THEM TRUE                                                                 141
XZ*****THOU ART MY LIFE,MY LOVE,MY HEART,THE VERY EYES OF ME                     142
---------------------------------------------------------------------------------143
+++++THIS PS FUNCTION HAS NULL STRING+++++                                        144
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                    145
                                                                                 146
                                                                                 147
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                    148
---------------------------------------------------------------------------------149
**************TEST-8  FUNCTIONS  RS*PS*CL*ES*SS**************************        150
+++++NUMBER OF ARGUMENTS LESS THAN NO OF MARKERS                                 151
NOMARKERS-NOARG MARKERS REPLACED BY NULL+++++                                    152
```

```
+++++NUMBER OF ARGUMENTS LESS THAN NO OF MARKERS                          153
NOMARKERS-NOARG MARKERS REPLACED BY NULL+++++                            154
XY*****THOUGH  SEAS AND LAND BETWIXT  US BOTH, OUR FAITH AND TROTH       155
XZ*****LIKE SEPARATED SOULS,ALL TIME AND SPACE CONTROLS                  156
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                        157
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++          158
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++          159
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++          160
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++          161
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++          162
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++          163
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++          164
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++          165
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++          166
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++          167
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++          168
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                        169
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++          170
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++          171
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++          172
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++          173
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++          174
XY**GH  SEAS  L BETWIXT  US BO, R FAI  TRO   XZ***LIKE SERATED SOU, SCE CONTRO  175
------------------------------------------------------------------------------ 176
+++++THIS PS FUNCTION HAS NULL STRING+++++                                177
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                             178
                                                                         179
                                                                         180
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                             181
------------------------------------------------------------------------ 182
**************TEST-9  FUNCTIONS  RS*PS*CL*DS*SS************************** 183
+++++NUMBER OF ARGUMENTS LESS THAN NO OF MARKERS                          184
NOMARKERS-NOARG MARKERS REPLACED BY NULL+++++                            185
+++++NUMBER OF ARGUMENTS LESS THAN NO OF MARKERS                          186
NOMARKERS-NOARG MARKERS REPLACED BY NULL+++++                            187
+++++NUMBER OF ARGUMENTS LESS THAN NO OF MARKERS                          188
NOMARKERS-NOARG MARKERS REPLACED BY NULL+++++                            189
NULL NAME FORM********SLEEP,SLEEP AGAIN,MY  LYRE  FOR THOU CANST NEVER  TELL MY  190
```

```
 HUMBLE TALE                                                                  191
ZZZ***IN SOUNDS THAT WILL PREVAIL,NOR GENTLE THOUGHTS IN HER INSPIRE          192
------------------------------------------------------------------------------ 193
+++++THIS PS FUNCTION HAS NULL STRING+++++                                     194
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                  195
                                                                               196
                                                                               197
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                  198
------------------------------------------------------------------------------ 199
**************TEST-10 FUNCTIONS  RS*PS*CL*DS*SS**************************     200
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              201
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++               202
HENCE,ALL YOU 2,AS SHORT AS ARE 1                                              203
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              204
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++               205
QQQQ****£(PS,HENCE(,)ALL YOU 2(,)AS SHORT AS ARE 1)                            206
+++++THIS PS FUNCTION HAS NULL STRING+++++                                     207
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                  208
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              209
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++               210
HENCE,ALL YOU M2,AS SHORT AS ARE M1                                            211
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              212
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++               213
QQQQ****£(PS,HENCE(,)ALL YOU M2(,)AS SHORT AS ARE M1)                          214
+++++THIS PS FUNCTION HAS NULL STRING+++++                                     215
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                  216
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              217
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++               218
HENCE,ALL YOU MARKER2,AS SHORT AS ARE MARKER1                                  219
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              220
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++               221
QQQQ****£(PS,HENCE(,)ALL YOU MARKER2(,)AS SHORT AS ARE MARKER1)                222
+++++THIS PS FUNCTION HAS NULL STRING+++++                                     223
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                  224
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              225
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++               226
HENCE,ALL YOU VAIN DELIGHTS,AS SHORT AS ARE THE NIGHTS                         227
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              228
```

```
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
QQQQ*****£(PS,HENCE(,)ALL YOU VAIN DELIGHTS(,)AS SHORT AS ARE THE NIGHTS)
+++++THIS PS FUNCTION HAS NULL STRING+++++
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++
-----------------------------------------------------------------------
+++++THIS PS FUNCTION HAS NULL STRING+++++
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++


+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++
-----------------------------------------------------------------------
**************TEST-11 FUNCTIONS  RS*PS*CL*DS*SS**************************
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
HENCE,ALL YOU 2,AS SHORT AS ARE 1
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
QQQQ*****£(PS,HENCE(,)ALL YOU 2(,)AS SHORT AS ARE 1)£(PS,QQQQ*****££(CL,QQQQ))£(S
S,QQQQ,£(RS))£(DS,QQQQ,££(CL,QQQQ,£(RS)))£(CL,QQQQ)
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
HENCE,ALL YOU M2,AS SHORT AS ARE M1
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
QQQQ*****£(PS,HENCE(,)ALL YOU M2(,)AS SHORT AS ARE M1)£(PS,QQQQ*****££(CL,QQQQ))£
(SS,QQQQ,£(RS))£(DS,QQQQ,££(CL,QQQQ,£(RS)))£(CL,QQQQ)
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
HENCE,ALL YOU MARKER2,AS SHORT AS ARE MARKER1
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
QQQQ*****£(PS,HENCE(,)ALL YOU MARKER2(,)AS SHORT AS ARE MARKER1)£(PS,QQQQ*****££(
CL,QQQQ))£(SS,QQQQ,£(RS))£(DS,QQQQ,££(CL,QQQQ,£(RS)))£(CL,QQQQ)
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
HENCE,ALL YOU VAIN DELIGHTS,AS SHORT AS ARE THE NIGHTS
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
```

```
QQQQ****£(PS,HENCE(,)ALL YOU VAIN DELIGHTS(,)AS SHORT AS ARE THE NIGHTS)£(PS,QQ
QQ****££(CL,QQQQ))£(SS,QQQQ,£(RS))£(DS,QQQQ,££(CL,QQQQ,£(RS)))£(CL,QQQQ)
-----------------------------------------------------------------------
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++SS"QQQQ"


+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++
-----------------------------------------------------------------------
**************TEST-12 FUNCTIONS  RS*PS*CL*DS*SS**************************
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
ABC*****WELCOME(,)ABC3 ARMS AND ABC1(,)A SIGH THAT ABC2 MORTIFIES
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++
ABC MARKERS SUBSTITUTED BY NULL*****WELCOME(,) ARMS AND (,)A SIGH THAT  MORTIFI
ES
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
PROCEDURE*****£(PS,£(CL,£(RS)))
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
WELCOME,3 ARMS AND 1,A SIGH THAT 2 MORTIFIES
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
WELCOME,FOLDED ARMS AND FIXED EYES,A SIGH THAT PIERCING MORTIFIES
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++
ABC LAST CALL*******WELCOME(,) ARMS AND (,)A SIGH THAT  MORTIFIES
+++++THIS PS FUNCTION HAS NULL STRING+++++
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++
-----------------------------------------------------------------------
+++++THIS PS FUNCTION HAS NULL STRING+++++
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++
```

```
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                  305
--------------------------------------------------------------------------------306
**************TEST-13 FUNCTICNS  RS*PS*CL*DS*SS**************************** 307
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                               308
+++++FCRM WITH NC MARKERS ,ARGUMENTS CF CL FUNCTICN IGNCRED+++++                309
THIS IS A CAT                                                                  310
+++++THIS PS FUNCTICN HAS NULL STRING+++++                                     311
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                  312
+++++THIS PS FUNCTICN HAS NULL STRING+++++                                     313
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                  314
THIS IS A DOG                                                                  315
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                  316
THIS IS A MOUSE                                                                317
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                  318
THIS IS A SHEEP                                                                319
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                  320
THIS IS A COW                                                                  321
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                  322
THIS IS A HORSE                                                                323
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                  324
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                               325
+++++THE CL FUNC HAS NC ARGUMENTS MARKERS REPLACED BY NULL+++++                 326
THIS IS A                                                                      327
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                  328
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                               329
+++++FCRM WITH NC MARKERS ,ARGUMENTS CF CL FUNCTICN IGNCRED+++++                330
THIS IS A CAT                                                                  331
+++++THIS PS FUNCTICN HAS NULL STRING+++++                                     332
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                  333
+++++THIS PS FUNCTICN HAS NULL STRING+++++                                     334
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                  335
THIS IS A DOG                                                                  336
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                  337
THIS IS A MOUSE                                                                338
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                  339
THIS IS A SHEEP                                                                340
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                  341
THIS IS A COW                                                                  342
```

```
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                  343
THIS IS A HORSE                                                                344
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                  345
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                              346
+++++THE CL FUNC HAS NC ARGUMENTS MARKERS REPLACED BY NULL+++++                 347
THIS IS A                                                                      348
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                  349
--------------------------------------------------------------------------     350
+++++THIS PS FUNCTICN HAS NULL STRING+++++                                      351
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                  352
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                  353
--------------------------------------------------------------------------     354
**************TEST-14 FUNCTICNS  RS*PS*CL*DS*SS**************************     355
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                              356
+++++FCRM WITH NC MARKERS ,ARGUMENTS OF CL FUNCTICN IGNORED+++++               357
POEM*****£(DS,VERSEM,Y)                                                        358
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                              359
+++++THE CL FUNC HAS NC ARGUMENTS MARKERS REPLACED BY NULL+++++                 360
+++++THE CL FUNC HAS NC ARGUMENTS MARKERS REPLACED BY NULL+++++                 361
POEM*****£(DS,VERSE,)                                                          362
£(DS,VERSE1,TC ME CNE SAVAGE HUNTING SCENE(,)MY SCLE DELIGHT THE HEADLONG RACE) 363
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                              364
+++++FCRM WITH NC MARKERS ,ARGUMENTS OF CL FUNCTICN IGNORED+++++               365
TO ME CNE SAVAGE HUNTING SCENE,MY SCLE DELIGHT THE HEADLONG RACE               366
                                                                               367
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                  368
£(DS,VERSE2,AND FRANTIC HURRY OF THE CHASE TO START(,)PURSUE(,)AND BRING TO BAY 369
)                                                                              370
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTICN+++++                              371
+++++FCRM WITH NC MARKERS ,ARGUMENTS OF CL FUNCTICN IGNORED+++++               372
AND FRANTIC HURRY OF THE CHASE TC START,PURSUE,AND BRING TC BAY                 373
+++++THIS PS FUNCTICN HAS NULL STRING+++++                                      374
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                  375
--------------------------------------------------------------------------     376
+++++THIS PS FUNCTICN HAS NULL STRING+++++                                      377
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                  378
                                                                               379
                                                                               380
```

```
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++                                  381
--------------------------------------------------------------------------------382
***************TEST-15 FUNCTIONS  RS*PS*CL*CS*SS*****************************  383
+++++THIS PS FUNCTION HAS NULL STRING+++++                                      384
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++                                   385
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               386
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 387
HE GAVE US THIS ETERNAL SPRING WHICH HERE ENAMELS EVERYTHING,                   388
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++                                   389
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               390
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 391
HE GAVE US THIS ETERNAL SPRING WHICH HERE ENAMELS EVERYTHING,                   392
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++                                   393
GIVE NAME FOR TEXT                                                              394
GIVE THE TEXT                                                                   395
GIVE THE NAME OF TEXT YOU CALL                                                  396
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               397
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 398
HE GAVE US THIS ETERNAL SPRING WHICH HERE ENAMELS EVERYTHING,                   399
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++                                   400
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               401
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 402
GIVE NAME FOR TEXT                                                              403
GIVE THE TEXT                                                                   404
GIVE THE NAME OF TEXT YOU CALL                                                  405
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               406
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 407
AND SENDS THE FOWLS TO US IN CARE ON DAILY VISITS THROUGH THE AIR               408
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               409
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 410
GIVE NAME FOR TEXT                                                              411
GIVE THE TEXT                                                                   412
GIVE THE NAME OF TEXT YOU CALL                                                  413
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               414
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 415
HE HANGS IN SHADES THE ORANGE BRIGHT LIKE GOLDEN LAMPS IN A GREEN NIGHT         416
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               417
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 418
```

```
GIVE NAME FOR TEXT                                                              419
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               420
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 421
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               422
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 423
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               424
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 425
VERSE1**HE GAVE US THIS ETERNAL SPRING WHICH HERE ENAMELS EVERYTHING,    VERSE2 426
**AND SENDS THE FOWLS TO US IN CARE ON DAILY VISITS THROUGH THE AIR     VERSE3** 427
HE HANGS IN SHADES THE ORANGE BRIGHT LIKE GOLDEN LAMPS IN A GREEN NIGHT         428
GIVE THE TEXT                                                                   429
--------------------------------------------------------------------------      430
+++++THIS STRING HAS NULL VALUE+++++                                            431
+++++DS""                                                                       432
GIVE THE NAME OF TEXT YOU CALL                                                  433
                                                                                434
                                                                                435
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                   436
--------------------------------------------------------------------------      437
**************TEST-16 FUNCTIONS  RS*PS*CL*DS*SS**************************      438
+++++THIS PS FUNCTION HAS NULL STRING+++++                                      439
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                   440
+++++THIS PS FUNCTION HAS NULL STRING+++++                                      441
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                   442
+++++THIS PS FUNCTION HAS NULL STRING+++++                                      443
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                   444
+++++THIS PS FUNCTION HAS NULL STRING+++++                                      445
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                   446
+++++THIS PS FUNCTION HAS NULL STRING+++++                                      447
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                   448
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               449
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 450
GIVE NAME**********                                                             451
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               452
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 453
PROGRAM SHOWING THE USE OF A PROCEDURE                                          454
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               455
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 456
```

```
GIVE NAME**********                                                    457
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                      458
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++       459
THEN LET NOT WHAT I CANNOT HAVE                                        460
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                      461
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++       462
GIVE NAME**********                                                    463
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                      464
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++       465
MY CHEER OF MIND DESTROY                                               466
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                      467
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++       468
GIVE NAME**********                                                    469
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                      470
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++       471
WILST THUS I SING,I AM A KING,ALTHOUGH A POOR BLIND BOY                472
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                      473
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++       474
GIVE NAME**********                                                    475
------------------------------------------------------------------     476
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                      477
+++++THERE IS NO SUCH FORM IN FSTORE TO BE CALLED   +++++              478
+++++CL"                                                               479
+++++THIS PS FUNCTION HAS NULL STRING+++++                             480
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                      481
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++       482
GIVE NAME**********                                                    483
                                                                       484
                                                                       485
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                          486
------------------------------------------------------------------     487
**************TEST-17 FUNCTIONS  RS*PS*CL*DS*SS**********************     488
                    STORE  OF  FORMS                                   489
**********************************************************************  490
            X1  MR SMITH HAS A SEAT IN THE HOUSE OF COMMONS            491
            X2  THE CLASSIC ROUTE TO THE MIDNIGHT SUN                  492
            X3  AFTER LUNCH YOU CONTINUE BY TRAIN A ROOM IS RESERVED FOR YOU  493
            X4  HE WENT TO PARIS WITH THE INTENTION OF LEARNING FRENCH  494
```

```
THE DEFINITION OF FORMS BEGINS                                                  495
+++++ERROR FUNCTION ,COMMA IS MISSING AFTER FIRST ARGUMENT+++++                 496
+++++CLX1"TAYLOR"FLAT"CITY"LONDON                                               497
MR TAYLOR HAS A FLAT IN THE CITY OF LONDON                                      498
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               499
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                  500
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                  501
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                  502
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                  503
 CLASSIC  TO                                                                    504
THE CLASSIC ROUTE TO THE MIDNIGHT SUN                                           505
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               506
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                  507
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                  508
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                  509
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                  510
 CLASSIC  TO                                                                    511
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                512
AFTER LUNCH MARKER1 CONTINUE MARKER3 MARKER4 MARKER2 A ROOM IS RESERVED FOR MAR 513
KER1                                                                            514
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               515
+++++THERE IS NO SUCH FORM IN FSTORE TO BE CALLED   +++++                       516
+++++CL"X5                                                                     517
+++++THIS PS FUNCTION HAS NULL STRING+++++                                      518
+++++NUMBER OF ARGUMENTS LESS THAN NO OF MARKERS                                519
NOMARKERS-NOARG MARKERS REPLACED BY NULL+++++                                   520
+++++NUMBER OF ARGUMENTS LESS THAN NO OF MARKERS                                521
NOMARKERS-NOARG MARKERS REPLACED BY NULL+++++                                   522
AFTER LUNCH YOU CONTINUE   BY TRAIN A ROOM IS RESERVED FOR YOU                  523
HE WENT TO STOCKHOLM WITH THE INTENTION OF MEETING FRIENDS                      524
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               525
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                  526
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                  527
 IS MY                                                                          528
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               529
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                  530
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                  531
 IS MY                                                                          532
```

```
THIS IS MY BOOK                                                                 533
                                                                                534
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                   535
-------------------------------------------------------------                   536
+++++THIS PS FUNCTICN HAS NULL STRING+++++                                      537
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                   538
                                                                                539
                                                                                540
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                   541
-------------------------------------------------------------------------       542
**************TEST-18 FUNCTICNS  RS*PS*CL*DS*SS**************************       543
+++++THIS PS FUNCTICN HAS NULL STRING+++++                                      544
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                   545
+++++THIS PS FUNCTICN HAS NULL STRING+++++                                      546
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                   547
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                               548
+++++FCRM WITH NC MARKERS ,ARGUMENTS CF CL FUNCTICN IGNCRED+++++                549
WITH HEAVY M3 I CFTEN M1 MCURN MY M2 WCE                                        550
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                               551
+++++FCRM WITH NC MARKERS ,ARGUMENTS CF CL FUNCTICN IGNCRED+++++                552
WITH HEAVY MM3 I CFTEN MM1 MCURN MY MM2 WCE                                     553
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                               554
+++++FCRM WITH NC MARKERS ,ARGUMENTS CF CL FUNCTICN IGNCRED+++++                555
WITH HEAVY MARKER3 I CFTEN MARKER1 MCURN MY MARKER2 WCE                         556
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                               557
+++++FCRM WITH NC MARKERS ,ARGUMENTS CF CL FUNCTICN IGNCRED+++++                558
WITH HEAVY SIGHS I CFTEN HEAR YCU MCURN MY HAPLESS WCE                          559
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                               560
+++++FCRM WITH NC MARKERS ,ARGUMENTS CF CL FUNCTICN IGNCRED+++++                561
-------------------------------------------------------------------------       562
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                               563
+++++THERE IS NC SUCH FCRM IN FSTCRE TC BE CALLED   +++++                       564
+++++CL"                                                                        565
+++++THIS PS FUNCTICN HAS NULL STRING+++++                                      566
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                               567
+++++FCRM WITH NC MARKERS ,ARGUMENTS CF CL FUNCTICN IGNCRED+++++                568
                                                                                569
                                                                                570
```

```
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++                                  571
--------------------------------------------------------------------------------  572
***************TEST-19 FUNCTIONS  RS*PS*CL*DS*SS**************************  573
+++++THIS PS FUNCTION HAS NULL STRING+++++                                      574
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++                                  575
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              576
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++               577
£(DS,X,Y)                                                                      578
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              579
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                580
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                581
£(DS,,)                                                                        582
+++++THIS PS FUNCTION HAS NULL STRING+++++                                      583
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++                                  584
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              585
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++               586
£(DS,VERSE1,THE FOND COMPLAINT(,)MY SONG(,)DISPROVE)                           587
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              588
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++               589
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              590
+++++THERE IS NO SUCH FORM IN FSTORE TO BE CALLED   +++++                      591
+++++CL"VERSE1                                                                 592
+++++THIS PS FUNCTION HAS NULL STRING+++++                                      593
£(DS,VVV,WWW)                                                                  594
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              595
+++++THERE IS NO SUCH FORM IN FSTORE TO BE CALLED   +++++                      596
+++++CL"                                                                       597
+++++THIS PS FUNCTION HAS NULL STRING+++++                                      598
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              599
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++               600
£(DS,VERSE2,AND JUSTIFY THE LAWS OF JOVE)                                      601
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              602
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++               603
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              604
+++++THERE IS NO SUCH FORM IN FSTORE TO BE CALLED   +++++                      605
+++++CL"VERSE2                                                                 606
+++++THIS PS FUNCTION HAS NULL STRING+++++                                      607
--------------------------------------------------------------------------------  608
```

```
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                609
+++++THERE IS NO SUCH FORM IN FSTORE TO BE CALLED   +++++                        610
+++++CL"                                                                        611
+++++THIS PS FUNCTION HAS NULL STRING+++++                                       612
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                613
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 614
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                    615
------------------------------------------------------------------------        616
**********1 TEST FOR FUNCTIONS TN,TF **************************                617
PS"                                                                             618
+++++THIS PS FUNCTION HAS NULL STRING+++++                                       619
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                    620
RS                                                                              621
DS"1"£(PS,POOR SOUL(,)THE CENTRE OF MY SINFUL EARTH)£(SS,1,£(RS))£(DS,1,££(CL,1 622
,£(RS)))                                                                        623
CL"1                                                                            624
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               625
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 626
PS"POOR SOUL,THE CENTRE OF MY SINFUL EARTH                                      627
POOR SOUL,THE CENTRE OF MY SINFUL EARTH                                         628
RS                                                                              629
SS"1"SOUL"EARTH                                                                 630
RS                                                                              631
CL"1"EARTH"SOUL                                                                 632
DS"1"£(PS,POOR EARTH(,)THE CENTRE OF MY SINFUL SOUL)£(SS,1,£(RS))£(DS,1,££(CL,1 633
,£(RS)))                                                                        634
CL"1                                                                            635
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               636
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 637
PS"POOR EARTH,THE CENTRE OF MY SINFUL SOUL                                      638
POOR EARTH,THE CENTRE OF MY SINFUL SOUL                                         639
RS                                                                              640
TF                                                                              641
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               642
+++++SS"1"                                                                      643
------------------------------------------------------------------------        644
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               645
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 646
```

```
+++++THIS PS FUNCTION HAS NULL STRING+++++
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++
-------------------------------------------------------------------------
**********2 TEST FOR FUNCTIONS TN,TF**************************
PS"
+++++THIS PS FUNCTION HAS NULL STRING+++++
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++
RS
DS"BUILDLINE"1 AND HIS DOG
SS"BUILDLINE"1"
DS"THISTHING"ONE MAN
DS"WTM"1 WENT TO MOW ,
SS"WTM"1"
TF


+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++
 AND HIS DOG
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
ONE MAN
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++
 WENT TO MOW ,
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
WENT TO MOW A MEADOW.

+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++
-------------------------------------------------------------------
+++++THIS PS FUNCTION HAS NULL STRING+++++
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++
-------------------------------------------------------------------------
**********3 TEST FOR FUNCTIONS TN,TF**************************
```

```
DS""WHOSE M1 NOT HIS M2 ARE                                                    685
SS""M1"M2                                                                      686
RS                                                                             687
DS"X1"1 LONG TO 2 BACK(,)AND 3 THAT ANCIENT TRACK                              688
RS                                                                             689
DS"X2"MARKER1 REACH THAT PLAIN(,) MARKER2 I LEFT MY MARKER3                    690
RS                                                                             691
DS"X3"BUT AH MARKER1 MARKER3 WITH MARKER2 STAY MARKER4                         692
RS                                                                             693
DS"X4"1MM THIS DUST FALLS 2MM(,)IN THAT 3MM(,)RETURN                           694
SS"X2"MARKER1"MARKER2"MARKER3                                                  695
SS"X4"1MM"2MM"3MM                                                              696
SS"X1"1"2"3                                                                    697
CL"                                                                            698
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              699
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                 700
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                 701
PS"WHOSE  NOT HIS  ARE                                                         702
WHOSE  NOT HIS  ARE                                                            703
CL"X1"O HOW I"TRAVEL"TREAD AGAIN                                               704
PS"O HOW I LONG TO TRAVEL BACK,AND TREAD AGAIN THAT ANCIENT TRACK              705
O HOW I LONG TO TRAVEL BACK,AND TREAD AGAIN THAT ANCIENT TRACK                 706
CL"X2"THAT I MIGHT ONCE MORE"WHERE FIRST"GLORIOUS TRAIN                        707
PS"THAT I MIGHT ONCE MORE REACH THAT PLAIN, WHERE FIRST I LEFT MY GLORIOUS TRAI 708
N                                                                              709
THAT I MIGHT ONCE MORE REACH THAT PLAIN, WHERE FIRST I LEFT MY GLORIOUS TRAIN  710
CL"X3"MY SOUL"TOO MUCH                                                         711
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++               712
PS"BUT AH MARKER1 MARKER3 WITH MARKER2 STAY MARKER4                            713
BUT AH MARKER1 MARKER3 WITH MARKER2 STAY MARKER4                               714
SS"X3"MARKER1"MARKER2"MARKER3"MARKER4                                          715
CL"X3"MY SOUL"TOO MUCH                                                         716
+++++NUMBER OF ARGUMENTS LESS THAN NO OF MARKERS                               717
NOMARKERS-NOARG MARKERS REPLACED BY NULL+++++                                  718
+++++NUMBER OF ARGUMENTS LESS THAN NO OF MARKERS                               719
NOMARKERS-NOARG MARKERS REPLACED BY NULL+++++                                  720
PS"BUT AH MY SOUL  WITH TOO MUCH STAY                                          721
BUT AH MY SOUL  WITH TOO MUCH STAY                                             722
```

```
CL"X4"AND WHEN"TC THE URN"STATE I COME                                          723
PS"AND WHEN THIS DUST FALLS TC THE URN,IN THAT STATE I COME,RETURN              724
AND WHEN THIS DUST FALLS TC THE URN,IN THAT STATE I COME,RETURN                 725
CL""PASSIONS"MASTERS                                                            726
PS"WHOSE PASSIONS NCT HIS MASTERS ARE                                           727
WHOSE PASSIONS NCT HIS MASTERS ARE                                              728
TF                                                                              729
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                               730
+++++THE CL FUNC HAS NC ARGUMENTS MARKERS REPLACED BY NULL+++++                 731
+++++THE CL FUNC HAS NC ARGUMENTS MARKERS REPLACED BY NULL+++++                 732
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTICN+++++                               733
+++++THE CL FUNC HAS NC ARGUMENTS MARKERS REPLACED BY NULL+++++                 734
+++++THE CL FUNC HAS NC ARGUMENTS MARKERS REPLACED BY NULL+++++                 735
+++++THE CL FUNC HAS NC ARGUMENTS MARKERS REPLACED BY NULL+++++                 736
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                               737
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                 738
+++++THE CL FUNC HAS NC ARGUMENTS MARKERS REPLACED BY NULL+++++                 739
+++++THE CL FUNC HAS NC ARGUMENTS MARKERS REPLACED BY NULL+++++                 740
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                               741
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                 742
+++++THE CL FUNC HAS NC ARGUMENTS MARKERS REPLACED BY NULL+++++                 743
+++++THE CL FUNC HAS NC ARGUMENTS MARKERS REPLACED BY NULL+++++                 744
+++++THE CL FUNC HAS NC ARGUMENTS MARKERS REPLACED BY NULL+++++                 745
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                               746
+++++THE CL FUNC HAS NC ARGUMENTS MARKERS REPLACED BY NULL+++++                 747
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                 748
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                 749
WHOSE  NOT HIS  ARE*** LCNG TC  BACK,ANC  THAT ANCIENT TRACK*** REACH THAT PLAI 750
N,  I LEFT MY ***BUT AH   WITH  STAY *** THIS DUST FALLS ,IN THAT ,RETURN       751
-------------------------------------------------------------------------------- 752
+++++THIS PS FUNCTION HAS NULL STRING+++++                                       753
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                   754
                                                                                755
                                                                                756
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                   757
-------------------------------------------------------------------------------- 758
**********4 TEST FCR FUNCTICNS TN,TF**********                                  759
SS""MARKER1"MARKER2"MARKER3"MARKER4"MARKER5                                     760
```

```
CL""EAR"IN"FORWARD
+++++NUMBER OF ARGUMENTS LESS THAN NO OF MARKERS
NOMARKERS-NOARG MARKERS REPLACED BY NULL+++++
+++++NUMBER OF ARGUMENTS LESS THAN NO OF MARKERS
NOMARKERS-NOARG MARKERS REPLACED BY NULL+++++
+++++NUMBER OF ARGUMENTS LESS THAN NO OF MARKERS
NOMARKERS-NOARG MARKERS REPLACED BY NULL+++++
+++++NUMBER OF ARGUMENTS LESS THAN NO OF MARKERS
NOMARKERS-NOARG MARKERS REPLACED BY NULL+++++
+++++NUMBER OF ARGUMENTS LESS THAN NO OF MARKERS
NOMARKERS-NOARG MARKERS REPLACED BY NULL+++++
+++++NUMBER OF ARGUMENTS LESS THAN NO OF MARKERS
NOMARKERS-NOARG MARKERS REPLACED BY NULL+++++
+++++NUMBER OF ARGUMENTS LESS THAN NO OF MARKERS
NOMARKERS-NOARG MARKERS REPLACED BY NULL+++++
PS"THE FORWARD YOUTH THAT WOULD APPEAR,MUST  NOW FORSAKE HIS  MUSES DEAR,  NOR
IN THE SHADOWS SING  HIS  NUMBERS LANGUISHING.
THE FORWARD YOUTH THAT WOULD APPEAR,MUST  NOW FORSAKE HIS  MUSES DEAR,  NOR IN
THE SHADOWS SING  HIS  NUMBERS LANGUISHING.
RS
DS"XXX"1 TIME 2 LEAVE 3 BOOKS 4 DUST
SS"XXX"1"2"3"4
TF
TIS TIME TO LEAVE THE BOOKS IN DUST
--------------------------------------------------------------------------------
"
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++


+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++
--------------------------------------------------------------------------------
**********5 TEST FOR FUNCTIONS TN,TF **************************
RS
DS"DD"T89NK 80W MANY ROYAL BONES SLEEP W9T89N T8ESE 8EARS O6 STONES
SS"CC"1"2"3"4"5
SS"DD"6"7"8"9
TF
FIRST VERSE  **MORTALITY,BEHOLD AND FEAR WHAT A CHANGE OF FLESH IS HERE
```

```
SECOND VERSE **THINK HCW MANY ROYAL BCNES SLEEP WITHIN THESE HEARS CF STCNES
--------------------------------------------------------------------------
+++++THIS PS FUNCTICN HAS NULL STRING+++++
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++


+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++
--------------------------------------------------------------------------
**********6 TEST FCR TN,TF  FUNCTICNS**********
DS"AA"1A(,)CR 1B(,)CR 1C IN ARMS(,)  WHCSE CHANCE CN THESE DEFENCELESS DOORS MA
Y SEIZE
SS"AA"1A"1B"1C
DS"PROG"£(PS,£(CL,£(RS)))
TF
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++
+++++FCRM WITH NC MARKERS ,ARGUMENTS CF CL FUNCTION IGNORED+++++
CAPTAIN,OR COLONEL,OR KNIGHT IN ARMS,  WHCSE CHANCE CN THESE DEFENCELESS DCORS
MAY SEIZE
--------------------------------------------------------------------------
+++++THIS PS FUNCTICN HAS NULL STRING+++++
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++


+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++
--------------------------------------------------------------------------
**********7 TEST FCR TN,TF  FUNCTICNS**********
DS"ABAB"TC SAVE GREEK WALLS FRCM RUIN BARE.
SS"ABAB"GREEK
DS"BABA"£(CL,ABAB,ATHENIAN)
PS"£(CL,BABA)
£(CL,BABA)
CL"BABA
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++
+++++FCRM WITH NC MARKERS ,ARGUMENTS CF CL FUNCTICN IGNORED+++++
PS"£(CL,ABAB,ATHENIAN)
£(CL,ABAB,ATHENIAN)
CL"BABA
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++
```

```
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
CL"ABAB"ATHENIAN
TF
TO SAVE ATHENIAN WALLS FROM RUIN BARE.
--------------------------------------------------------------------------
""""
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++


+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++
--------------------------------------------------------------------------
**********8 TEST FOR FUNCTIONS TN,TF **********
DS"SONG"SOME THING WENT TO MOW WENT TO MOW A MEADOW ELSE AND HIS DOG WENT TO MO
W A MEADOW
SS"SONG"SOME"THING"ELSE
CL"SONG"ONE"MAN"ONE MAN
PS"ONE MAN WENT TO MOW WENT TO MOW A MEADOW ONE MAN AND HIS DOG WENT TO MOW A M
EADOW
ONE MAN WENT TO MOW WENT TO MOW A MEADOW ONE MAN AND HIS DOG WENT TO MOW A MEAD
OW
TF
TWO MEN WENT TO MOW WENT TO MOW A MEADOW TWO MEN  ONE MAN AND HIS DOG WENT TO
MOW A MEADOW
--------------------------------------------------------------------------
"                                                                      ""
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++
--------------------------------------------------------------------------
**********1  TEST FOR FUNCTION DA **********
ONE MAN WENT TO MOW WENT TO MOW A MEADOW ONE MAN AND HIS DOG WENT TO MOW A MEAD
OW
+++++THERE IS NO SUCH FORM IN FSTORE TO BE CALLED  +++++
+++++CL"SONG"TWO"MEN"TWO MEN  ONE MAN
+++++THIS PS FUNCTION HAS NULL STRING+++++
""""
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++
```

```
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++                                  875
--------------------------------------------------------------------------------876
**********2  TEST FOR FUNCTION DA ********************************************877
£(CL,B)                                                                        878
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              879
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++               880
£(CL,A,DEAD)                                                                   881
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              882
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++               883
NO LONGER MOURN FOR ME WHEN I AM DEAD                                          884
NAME  A                                                                        885
NAME  B                                                                        886
+++++THERE ARE NO FORMS IN STORE+++++                                          887
+++++THERE ARE NO FORMS IN STORE+++++                                          888
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              889
+++++THERE IS NO SUCH FORM IN FSTORE TO BE CALLED   +++++                      890
+++++CL"B                                                                      891
+++++THIS PS FUNCTION HAS NULL STRING+++++                                     892
""""                                                                           893
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++                                  894
                                                                               895
                                                                               896
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++                                  897
--------------------------------------------------------------------------------898
**********3  TEST FOR FUNCTION DA ********************************************899
LEST THE WISE WORLD SHOULD LOOK INTO YOUR MOAN                                 900
AND MOCK YOU WITH ME AFTER I AM GONE.                                          901
*****VERSE1                                                                    902
*****VERSE2                                                                    903
+++++THERE ARE NO FORMS IN STORE+++++                                          904
+++++THERE ARE NO FORMS IN STORE+++++                                          905
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              906
+++++THERE IS NO SUCH FORM IN FSTORE TO BE CALLED   +++++                      907
+++++CL"VERSE1                                                                 908
+++++THIS PS FUNCTION HAS NULL STRING+++++                                     909
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              910
+++++THERE IS NO SUCH FORM IN FSTORE TO BE CALLED   +++++                      911
+++++CL"VERSE2                                                                 912
```

```
+++++THIS PS FUNCTION HAS NULL STRING+++++                                913
"""                                                                      914
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                            915
                                                                         916
                                                                         917
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                            918
------------------------------------------------------------------------ 919
*********************4  TEST FOR FUNCTION DA ****************************** 920
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                        921
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++         922
THROUGH DISCONTENT OF MY LONG FRUITLESS STAY                             923
*****A1                                                                  924
*****PROG                                                                925
+++++THERE ARE NO FORMS IN STORE+++++                                    926
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                        927
+++++THERE IS NO SUCH FORM IN FSTORE TO BE CALLED   +++++                928
+++++CL"PROG                                                            929
"""                                                                      930
                                                                         931
                                                                         932
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                            933
                                                                         934
                                                                         935
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                            936
------------------------------------------------------------------------ 937
*********************5  TEST FOR FUNCTION DA ****************************** 938
WITH THAT I SAW TWO SWANS OF GOODLY HUE,COME SOFTLY SWIMMING DOWN ALONG THE LEE  939
TWO FAIRER BIRDS I YET DID NEVER SEE              ,                      940
*****                                                                    941
*****XZ                                                                  942
+++++THERE ARE NO FORMS IN STORE+++++                                    943
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                        944
+++++THERE IS NO SUCH FORM IN FSTORE TO BE CALLED   +++++                945
+++++CL"                                                                946
+++++THIS PS FUNCTION HAS NULL STRING+++++                               947
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                        948
+++++THERE IS NO SUCH FORM IN FSTORE TO BE CALLED   +++++                949
+++++CL"XZ                                                               950
```

```
+++++THIS PS FUNCTION HAS NULL STRING+++++                                951
""                                                                       952
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++                            953
                                                                         954
                                                                         955
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++                            956
------------------------------------------------------------------------ 957
**********************6  TEST FOR FUNCTION DA ***************************** 958
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                        959
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++          960
 AND HIS DOG                                                             961
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                        962
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++         963
ONE MAN                                                                  964
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                        965
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++          966
 WENT TO MOW ,                                                           967
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                        968
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++         969
WENT TO MOW A MEADOW.                                                    970
*****BUILDLINE                                                           971
*****THISTHING                                                           972
*****WTM                                                                 973
*****WTMAN                                                               974
+++++THERE ARE NO FORMS IN STORE+++++                                    975
+++++THERE ARE NO FORMS IN STORE+++++                                    976
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                        977
+++++THERE IS NO SUCH FORM IN FSTORE TO BE CALLED   +++++                978
+++++CL"BUILDLINE                                                        979
+++++THIS PS FUNCTION HAS NULL STRING+++++                               980
                                                                         981
                                                                         982
                                                                         983
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++                            984
                                                                         985
                                                                         986
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++                            987
------------------------------------------------------------------------ 988
```

```
**********7  TEST FOR FUNCTION DA ******************************          989
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                          990
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++           991
LET ENDLESS PEACE YOUR STEADFAST HEARTS ACCORD                             992
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                          993
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++           994
LET ENDLESS WAR YOUR STEADFAST HEARTS ACCORD                               995
*****QQ                                                                    996
+++++THERE ARE NO FORMS IN STORE+++++                                      997
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                          998
+++++THERE IS NO SUCH FORM IN FSTORE TO BE CALLED   +++++                  999
+++++CL"QQ                                                                1000
+++++THIS PS FUNCTION HAS NULL STRING+++++                                 1001
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                              1002
                                                                          1003
                                                                          1004
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                              1005
--------------------------------------------------------------------------1006
**********8  TEST FOR FUNCTION DA ******************************          1007
                    STORE  OF  FORMS                                      1008
**************************************************************************1009
             X1  MR SMITH HAS A SEAT IN THE HOUSE OF COMMONS              1010
             X2  THE CLASSIC ROUTE TO THE MIDNIGHT SUN                    1011
             X3  AFTER LUNCH YOU CONTINUE BY TRAIN A ROOM IS RESERVED FOR YOU  1012
             X4  HE WENT TO PARIS WITH THE INTENTION OF LEARNING FRENCH    1013
THE DEFINITION OF FORMS BEGINS                                             1014
+++++ERROR FUNCTION ,COMMA IS MISSING AFTER FIRST ARGUMENT+++++           1015
+++++CLX1"TAYLOR"FLAT"CITY"LONDON                                         1016
MR TAYLOR HAS A FLAT IN THE CITY OF LONDON                                 1017
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                          1018
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++            1019
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++            1020
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++            1021
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++            1022
 CLASSIC  TO                                                               1023
THE CLASSIC ROUTE TO THE MIDNIGHT SUN                                      1024
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                          1025
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++            1026
```

```
+++++THE CL FUNC HAS NC ARGUMENTS MARKERS REPLACED BY NULL+++++                  1027
+++++THE CL FUNC HAS NC ARGUMENTS MARKERS REPLACED BY NULL+++++                  1028
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                  1029
 CLASSIC  TO                                                                    1030
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1031
AFTER LUNCH MARKER1 CONTINUE MARKER3 MARKER4 MARKER2 A ROOM IS RESERVED FOR MAR 1032
KER1                                                                           1033
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTICN+++++                               1034
+++++THERE IS NO SUCH FORM IN FSTORE TO BE CALLED   +++++                       1035
+++++CL"X5                                                                     1036
+++++THIS PS FUNCTICN HAS NULL STRING+++++                                      1037
+++++NUMBER OF ARGUMENTS LESS THAN NC OF MARKERS                                1038
NOMARKERS-NOARG MARKERS REPLACED BY NULL+++++                                  1039
+++++NUMBER OF ARGUMENTS LESS THAN NC OF MARKERS                                1040
NOMARKERS-NOARG MARKERS REPLACED BY NULL+++++                                  1041
AFTER LUNCH YOU CONTINUE   BY TRAIN A ROOM IS RESERVED FOR YOU                 1042
HE WENT TO STOCKHOLM WITH THE INTENTION OF MEETING FRIENDS                     1043
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                               1044
+++++THE CL FUNC HAS NC ARGUMENTS MARKERS REPLACED BY NULL+++++                 1045
+++++THE CL FUNC HAS NC ARGUMENTS MARKERS REPLACED BY NULL+++++                 1046
 IS MY                                                                         1047
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                               1048
+++++THE CL FUNC HAS NC ARGUMENTS MARKERS REPLACED BY NULL+++++                 1049
+++++THE CL FUNC HAS NC ARGUMENTS MARKERS REPLACED BY NULL+++++                 1050
 IS MY                                                                         1051
THIS IS MY BCCK                                                                1052
                                                                               1053
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                   1054
*****                                                                          1055
*****X2                                                                        1056
*****X4                                                                        1057
*****X1                                                                        1058
*****X3                                                                        1059
+++++THERE ARE NC FCRMS IN STORE+++++                                          1060
+++++THERE ARE NC FCRMS IN STORE+++++                                          1061
+++++THIS PS FUNCTICN HAS NULL STRING+++++                                      1062
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                   1063
                                                                               1064
```

```
                                                                                1065
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                   1066
--------------------------------------------------------------------------------1067
********************TEST-1 LN FUNCTICN*****************************************1068
+++++THIS PS FUNCTICN HAS NULL STRING+++++                                       1069
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                   1070
+++++THERE ARE NC FORMS IN STORE+++++                                           1071
+++++THERE ARE NC FORMS IN STCRE+++++                                           1072
+++++ERRCR FUNCTICN,COMMA IS MISSING AFTER FIRST ARGUMENT+++++                   1073
+++++LN COMMA MISSING                                                           1074
+++++THERE ARE NC FORMS IN STORE+++++                                           1075
+++++THIS PS FUNCTICN HAS NULL STRING+++++                                       1076
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++                                   1077
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                               1078
+++++FCRM WITH NC MARKERS ,ARGUMENTS CF CL FUNCTICN IGNCRED+++++                1079
YOUTH IS FULL CF TRCUBLES AGE IS FULL CF CARE                                   1080
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                               1081
+++++FCRM WITH NC MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                1082
YOUTH IS FULL CF PLEASANCE AGE IS FULL CF CARE                                  1083
NAMEVVV                                                                         1084
+++++THERE ARE NC FCRMS IN STCRE+++++                                          1085
NAME  VVV                                                                       1086
+++++THERE ARE NC FCRMS IN STCRE+++++                                          1087
+++++ERRCR FUNCTICN,COMMA IS MISSING AFTER FIRST ARGUMENT+++++                  1088
+++++LN COMMA MISSING                                                          1089
VVV                                                                             1090
+++++THERE ARE NC FORMS IN STORE+++++                                          1091
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTICN+++++                              1092
+++++SS"VVV"                                                                   1093
--------------------------------------------------------------------------------1094
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                               1095
+++++FCRM WITH NC MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                1096
+++++THIS PS FUNCTICN HAS NULL STRING+++++                                       1097
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                   1098
                                                                                1099
                                                                                1100
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                   1101
--------------------------------------------------------------------------------1102
```

```
**********TEST-2 LN FUNCTION**************************************  1103
+++++THIS PS FUNCTION HAS NULL STRING+++++                          1104
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                       1105
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTION+++++                   1106
+++++THE CL FUNC HAS NC ARGUMENTS MARKERS REPLACED BY NULL+++++     1107
 AND HIS DCG                                                       1108
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                   1109
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++    1110
ONE MAN                                                            1111
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTION+++++                   1112
+++++THE CL FUNC HAS NC ARGUMENTS MARKERS REPLACED BY NULL+++++     1113
 WENT TO MOW ,                                                     1114
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTION+++++                   1115
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++    1116
WENT TC MOW A MEADOW.                                              1117
                                                                   1118
                                                                   1119
                                                                   1120
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                       1121
NAMEBUILDLINE                                                      1122
NAMETHISTHING                                                      1123
NAMEWTM                                                            1124
NAMEWTMAM                                                          1125
+++++THERE ARE NO FORMS IN STORE+++++                              1126
NAME  BUILDLINE                                                    1127
NAME  THISTHING                                                    1128
NAME  WTM                                                          1129
NAME  WTMAM                                                        1130
+++++THERE ARE NC FORMS IN STORE+++++                              1131
+++++ERROR FUNCTION,COMMA IS MISSING AFTER FIRST ARGUMENT+++++     1132
+++++LN COMMA MISSING                                             1133
BUILDLINE                                                          1134
THISTHING                                                          1135
WTM                                                                1136
WTMAM                                                              1137
+++++THERE ARE NO FORMS IN STORE+++++                              1138
+++++THIS PS FUNCTION HAS NULL STRING+++++                         1139
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                      1140
```

```
                                                                                1141
                                                                                1142
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++                                   1143
--------------------------------------------------------------------------------1144
********************TEST-3 LN FUNCTION*****************************************1145
+++++THIS PS FUNCTION HAS NULL STRING+++++                                       1146
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++                                   1147
                     STORE  OF  FORMS                                           1148
********************************************************************************1149
              X1  MR SMITH HAS A SEAT IN THE HOUSE OF COMMONS                   1150
              X2  THE CLASSIC ROUTE TO THE MIDNIGHT SUN                         1151
              X3  AFTER LUNCH YOU CONTINUE BY TRAIN A ROOM IS RESERVED FOR YOU  1152
              X4  HE WENT TO PARIS WITH THE INTENTION OF LEARNING FRENCH        1153
THE DEFINITION OF FORMS BEGINS                                                  1154
+++++ERROR FUNCTION ,COMMA IS MISSING AFTER FIRST ARGUMENT+++++                 1155
+++++CLX1"TAYLOR"FLAT"CITY"LONDON                                               1156
MR TAYLOR HAS A FLAT IN THE CITY OF LONDON                                      1157
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               1158
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                  1159
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                  1160
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                  1161
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                  1162
 CLASSIC  TO                                                                    1163
THE CLASSIC ROUTE TO THE MIDNIGHT SUN                                           1164
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               1165
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                  1166
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                  1167
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                  1168
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++                  1169
 CLASSIC  TO                                                                    1170
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1171
AFTER LUNCH MARKER1 CONTINUE MARKER3 MARKER4 MARKER2 A ROOM IS RESERVED FOR MAR 1172
KER1                                                                            1173
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               1174
+++++THERE IS NO SUCH FORM IN FSTORE TO BE CALLED   +++++                       1175
+++++CL"X5                                                                      1176
+++++THERE IS NO SUCH FORM IN FSTORE TO BE CALLED   +++++                       1177
+++++CL"X5                                                                      1178
```

```
+++++THIS PS FUNCTION HAS NULL STRING+++++                                1179
+++++NUMBER OF ARGUMENTS LESS THAN NO OF MARKERS                          1180
NOMARKERS-NOARG MARKERS REPLACED BY NULL+++++                            1181
+++++NUMBER OF ARGUMENTS LESS THAN NO OF MARKERS                          1182
NOMARKERS-NOARG MARKERS REPLACED BY NULL+++++                            1183
AFTER LUNCH YOU CONTINUE   BY TRAIN A ROOM IS RESERVED FOR YOU           1184
HE WENT TO STOCKHOLM WITH THE INTENTION OF MEETING FRIENDS               1185
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                        1186
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++          1187
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++          1188
 IS MY                                                                   1189
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                        1190
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++          1191
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++          1192
 IS MY                                                                   1193
THIS IS MY BOOK                                                          1194
                                                                         1195
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                            1196
NAME                                                                     1197
NAMEX2                                                                   1198
NAMEX4                                                                   1199
NAMEX1                                                                   1200
NAMEX3                                                                   1201
+++++THERE ARE NO FORMS IN STORE+++++                                    1202
NAME                                                                     1203
NAME   X2                                                                1204
NAME   X4                                                                1205
NAME   X1                                                                1206
NAME   X3                                                                1207
+++++THERE ARE NO FORMS IN STORE+++++                                    1208
+++++ERROR FUNCTION,COMMA IS MISSING AFTER FIRST ARGUMENT+++++           1209
+++++LN COMMA MISSING                                                    1210
X2                                                                       1211
X4                                                                       1212
X1                                                                       1213
X3                                                                       1214
+++++THERE ARE NO FORMS IN STORE+++++                                    1215
+++++THIS PS FUNCTION HAS NULL STRING+++++                               1216
```

```
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++                                1217
                                                                             1218
                                                                             1219
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++                                1220
------------------------------------------------------------------------     1221
**********TEST-4 LN FUNCTION**************************************          1222
+++++THIS PS FUNCTION HAS NULL STRING+++++                                   1223
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++                                1224
+++++NUMBER OF ARGUMENTS LESS THAN NO OF MARKERS                            1225
NOMARKERS-NOARG MARKERS REPLACED BY NULL+++++                               1226
+++++NUMBER OF ARGUMENTS LESS THAN NO OF MARKERS                            1227
NOMARKERS-NOARG MARKERS REPLACED BY NULL+++++                               1228
+++++NUMBER OF ARGUMENTS LESS THAN NO OF MARKERS                            1229
NOMARKERS-NOARG MARKERS REPLACED BY NULL+++++                               1230
+++++NUMBER OF ARGUMENTS LESS THAN NO OF MARKERS                            1231
NOMARKERS-NOARG MARKERS REPLACED BY NULL+++++                               1232
WHEN BIRDS DO  SING  HEY DING A DING DING,SWEET LOVERS LOVE THE  SPRING .   1233
SO TRUE A FOOL IS LOVE,THAT IN YOUR WILL,THOUGH YOU DO ANYTHING,HE THINKS NO 1234
ILL. W.SHAKSPEARE                                                           1235
""                                                                          1236
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++                                1237
NAME                                                                        1238
NAMEXZ                                                                      1239
+++++THERE ARE NO FORMS IN STORE+++++                                       1240
NAME                                                                        1241
NAME  XZ                                                                    1242
+++++THERE ARE NO FORMS IN STORE+++++                                       1243
+++++ERROR FUNCTION,COMMA IS MISSING AFTER FIRST ARGUMENT+++++              1244
+++++LN COMMA MISSING                                                       1245
XZ                                                                          1246
+++++THERE ARE NO FORMS IN STORE+++++                                       1247
+++++THIS PS FUNCTION HAS NULL STRING+++++                                   1248
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++                                1249
------------------------------------------------------------------------     1250
+++++THIS PS FUNCTION HAS NULL STRING+++++                                   1251
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++                                1252
                                                                            1253
                                                                            1254
```

```
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                  1255
-------------------------------------------------------------------------------1256
**********TEST-5 LN FUNCTICN**************************************************1257
+++++THIS PS FUNCTICN HAS NULL STRING+++++                                      1258
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                  1259
SO LONG AS MEN CAN BREATHE OR EYES CAN SEE, SO LONG LIVES THIS, AND THIS GIVES  1260
LIFE TC THEE.                                                                  1261
FORGET NCT YET THE  TRIED INTENT  OF SUCH A  TRUTH AS I HAVE MEANT , MY  GREAT  1262
TRAVAIL SC GLADLY SPENT,FORGET NCT YET.                                        1263
+++++THIS PS FUNCTICN HAS NULL STRING+++++                                      1264
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                  1265
NAMEXYZ                                                                        1266
NAMEXYW                                                                        1267
+++++THERE ARE NC FCRMS IN STCRE+++++                                          1268
NAME  XYZ                                                                      1269
NAME  XYW                                                                      1270
+++++THERE ARE NC FCRMS IN STCRE+++++                                          1271
+++++ERRCR FUNCTICN,CCMMA IS MISSING AFTER FIRST ARGUMENT+++++                  1272
+++++LN COMMA MISSING                                                          1273
XYZ                                                                            1274
XYW                                                                            1275
+++++THERE ARE NC FCRMS IN STCRE+++++                                          1276
+++++THIS PS FUNCTICN HAS NULL STRING+++++                                      1277
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                  1278
-------------------------------------------------------------------------------1279
+++++THIS PS FUNCTICN HAS NULL STRING+++++                                      1280
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                  1281
                                                                               1282
                                                                               1283
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                  1284
-------------------------------------------------------------------------------1285
**********TEST-6 LN FUNCTICN**************************************************1286
+++++THIS PS FUNCTICN HAS NULL STRING+++++                                      1287
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                  1288
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                              1289
+++++THE CL FUNC HAS NC ARGUMENTS MARKERS REPLACED BY NULL+++++                1290
+++++THE CL FUNC HAS NC ARGUMENTS MARKERS REPLACED BY NULL+++++                1291
+++++THE CL FUNC HAS NC ARGUMENTS MARKERS REPLACED BY NULL+++++                1292
```

```
ART THOU (,)YET (,)C .                                                    1293
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                         1294
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++          1295
ART THOU RICH,YET IS THY MIND PERPLEXED,C PUNISHMENT.                     1296
NAMEA1                                                                    1297
NAMEPROG                                                                  1298
+++++THERE ARE NO FORMS IN STORE+++++                                     1299
NAME  A1                                                                  1300
NAME  PROG                                                                1301
+++++THERE ARE NO FORMS IN STORE+++++                                     1302
+++++ERROR FUNCTION,COMMA IS MISSING AFTER FIRST ARGUMENT+++++            1303
+++++LN COMMA MISSING                                                     1304
A1                                                                        1305
PROG                                                                      1306
+++++THERE ARE NO FORMS IN STORE+++++                                     1307
+++++THIS PS FUNCTION HAS NULL STRING+++++                                1308
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                             1309
------------------------------------------------------------------------  1310
+++++THIS PS FUNCTION HAS NULL STRING+++++                                1311
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                             1312
                                                                          1313
                                                                          1314
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                             1315
------------------------------------------------------------------------  1316
**********TEST-7 LN FUNCTION**************************************       1317
+++++THIS PS FUNCTION HAS NULL STRING+++++                                1318
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                             1319
£(CL,B)                                                                   1320
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                         1321
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++          1322
£(CL,A,GIVE,LIVE)                                                         1323
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                         1324
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++          1325
THESE DELIGHTS IF THOU CANST GIVE,MIRTH,WITH THEE I MEAN TO LIVE.         1326
NAMEA                                                                     1327
NAMEB                                                                     1328
+++++THERE ARE NO FORMS IN STORE+++++                                     1329
NAME  A                                                                   1330
```

```
NAME  B                                                                    1331
+++++THERE ARE NO FORMS IN STORE+++++                                      1332
+++++ERROR FUNCTION,COMMA IS MISSING AFTER FIRST ARGUMENT+++++             1333
+++++LN COMMA MISSING                                                      1334
A                                                                          1335
B                                                                          1336
+++++THERE ARE NO FORMS IN STORE+++++                                      1337
-------------------------------------------------------------------------- 1338
-                                                                          1339
""""                                                                       1340
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++                              1341
---------------------------------------------------------------------      1342
+++++THIS PS FUNCTION HAS NULL STRING+++++                                  1343
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++                              1344
                                                                           1345
                                                                           1346
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++                              1347
---------------------------------------------------------------------      1348
**********TEST-8 LN FUNCTION**********************************           1349
+++++THIS PS FUNCTION HAS NULL STRING+++++                                  1350
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++                              1351
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                          1352
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++           1353
OW                                                                         1354
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                          1355
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++           1356
TWO MEN WENT TO MOW WENT TO MOW A MEADOW TWO MEN  ONE MAN AND HIS DOG WENT TO M 1357
OW A MEADOW                                                                1358
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                          1359
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++           1360
THREE MEN WENT TO MOW WENT TO MOW A MEADOW THREE MEN  TWO MEN  ONE MAN AND HIS  1361
DOG WENT TO MOW A MEADOW                                                   1362
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                          1363
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++           1364
NAMESONG                                                                   1365
NAMESING                                                                   1366
+++++THERE ARE NO FORMS IN STORE+++++                                      1367
NAME  SONG                                                                 1368
```

```
NAME  SING                                                                1369
+++++THERE ARE NO FORMS IN STORE+++++                                     1370
+++++ERROR FUNCTION,COMMA IS MISSING AFTER FIRST ARGUMENT+++++            1371
+++++LN COMMA MISSING                                                     1372
SONG                                                                      1373
SING                                                                      1374
+++++THERE ARE NO FORMS IN STORE+++++                                     1375
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                         1376
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++           1377
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++           1378
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++           1379
  WENT TO MOW WENT TO MOW A MEADOW  AND HIS DOG WENT TO MOW A MEADOW      1380
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                         1381
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++          1382
------------------------------------------------------------------------  1383
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                         1384
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++           1385
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++           1386
+++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NULL+++++           1387
  WENT TO MOW WENT TO MOW A MEADOW  AND HIS DOG WENT TO MOW A MEADOW      1388
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                         1389
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++          1390
                                                                          1391
                                                                          1392
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                            1393
------------------------------------------------------------------------  1394
**********TEST-1 PF FUNCTION**************************************       1395
+++++THIS PS FUNCTION HAS NULL STRING+++++                                1396
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                            1397
+++++ THERE IS NOT SUCH A FORM IN STORE+++++                             1398
+++++PF"NAME                                                             1399
IT HAS &2REFORE BEEN SUGGESTED &1 MARY WAS ACTUALLY BORN ON &2 7TH AND &1 &2 DA 1400
TE WAS ALTERED TO &2 8TH &3 ORDER TO CO&3CIDE WITH &2 FEAST OF &2 VIRG&3.       1401
+++++ THERE IS NOT SUCH A FORM IN STORE+++++                             1402
+++++PF"                                                                 1403
DURING THE WHOLE OF THIS PERIOD,HAMILTON BLOOD WAS GENERALLY CONSIDERED A CONVE 1404
NIENT SCAPEGOAT ON WHICH TO BLAME ABNORMALITIES OF TEMPER.               1405
DURING THE WHOLE &1 THIS PERIOD,HAMILTON BLOOD WAS GENERALLY CONSIDERED A CONVE 1406
```

```
NIENT SCAPEGCAT CN WHICH TO BLAME &2 &1 TEMPER.                                 1407
THIS STRING HAS LESS THAN 80 CHARACTERS                                         1408
&1 STRING HAS LESS THAN 80 &2                                                   1409
""                                                                              1410
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                   1411
                                                                                1412
                                                                                1413
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                   1414
--------------------------------------------------------------------------      1415
**********TEST-2 PF FUNCTICN**************************************              1416
+++++THIS PS FUNCTICN HAS NULL STRING+++++                                       1417
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                   1418
+++++ THERE IS NCT SUCH A FORM IN STORE+++++                                     1419
+++++PF"NAME                                                                    1420
+++++ THERE IS NCT SUCH A FORM IN STORE+++++                                     1421
+++++PF"TEXT                                                                    1422
IT HAS &2REFCRE BEEN SUGGESTED &1 MARY WAS ACTUALLY BORN CN &2 7TH AND &1 &2 DA 1423
TE WAS ALTERED TC &2 8TH &3 CRDER TO CO&3CIDE WITH &2 FEAST OF &2 VIRG&3.       1424
+++++ THERE IS NCT SUCH A FORM IN STORE+++++                                     1425
+++++PF"                                                                        1426
DURING THE WHCLE CF THIS PERICD,HAMILTCN BLCCD WAS GENERALLY CCNSIDERED A CCNVE 1427
NIENT SCAPEGCAT CN WHICH TC BLAME ABNCRMALITIES CF TEMPER.                      1428
DURING THE WHCLE &1 THIS PERICD,HAMILTCN BLCCD WAS GENERALLY CCNSIDERED A CCNVE 1429
NIENT SCAPEGCAT CN WHICH TC BLAME &2 &1 TEMPER.                                 1430
THIS STRING HAS LESS THAN 80 CHARACTERS                                         1431
&1 STRING HAS LESS THAN 80 &2                                                   1432
+++++ERROR FUNCTION ,COMMA IS MISSING AFTER FIRST ARGUMENT+++++                 1433
+++++PF AB                                                                      1434
&9T &8&1S &2R&5&6CR&5 &2&5&5&14 SU&7&7&5ST&5&4 &1 &13&1RY W&1S &1&3TU&1&12&12Y  1435
&2CR&14 C&14 &2 7T&8 &1&14&4 &1 &2 &4&1T&5 W&1S &1&12T&5R&5&4 TC &2 &T&& &3 CR& 1436
4&5R TC &3C&3&3&9&4&5 W&9T&8 &2 &6&5&1ST C&6 &2 V&9R&7&3.                       1437
&4URIN&7 TH&5 WHCL&5 &1 THIS P&5RIC&4,H&1MILTCN &2LCC&4 W&1S &7&5N&5R&1LLY &3CN 1438
SI&4&5R&5&4 &1 &3CNV&5NI&5NT S&3&1P&5&7C&1T CN WHI&3H TC &2L&1M&5 &2 &1 T&5MP&5 1439
R.                                                                              1440
""                                                                              1441
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                   1442
                                                                                1443
                                                                                1444
```

```
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                   1445
--------------------------------------------------------------------------------1446
**********TEST-3 PF FUNCTICN*****************************************          1447
+++++THIS PS FUNCTICN HAS NULL STRING+++++                                       1448
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                   1449
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                               1450
+++++FCRM WITH NC MARKERS ,ARGUMENTS CF CL FUNCTICN IGNORED+++++                 1451
SHE IS DESCRIBED AS VERY PRETTY INDEED                                          1452
£(PS,(SHE IS DESCRIBED AS &1))£(SS,QQ,££(RS))££(PF,QQ) £(DS,QQ,££(CL,QQ,£(RS))) 1453
£(PF,QQ)£(CL,QQ)                                                                1454
£(PS,(SHE IS DESCRIBED AS VERY CLEVER))£(SS,QQ,££(RS))££(PF,QQ) £(DS,QQ,££(CL,Q 1455
Q,£(RS)))£(PF,QQ)£(CL,QQ)                                                       1456
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                               1457
+++++FCRM WITH NC MARKERS ,ARGUMENTS CF CL FUNCTICN IGNORED+++++                 1458
SHE IS DESCRIBED AS VERY CLEVER                                                 1459
£(PS,(SHE IS DESCRIBED AS &1))£(SS,QQ,££(RS))££(PF,QQ) £(DS,QQ,££(CL,QQ,£(RS))) 1460
£(PF,QQ)£(CL,QQ)                                                                1461
£(PS,(SHE IS DESCRIBED AS AWFULLY BORING))£(SS,QQ,££(RS))££(PF,QQ) £(DS,QQ,££(C 1462
L,QQ,£(RS)))£(PF,QQ)£(CL,QQ)                                                    1463
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                               1464
+++++FCRM WITH NO MARKERS ,ARGUMENTS CF CL FUNCTICN IGNORED+++++                 1465
SHE IS DESCRIBED AS AWFULLY BORING                                              1466
                                                                                1467
                                                                                1468
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                   1469
--------------------------------------------------------------------------------1470
**********TEST-4 PF FUNCTICN*****************************************          1471
+++++THIS PS FUNCTICN HAS NULL STRING+++++                                       1472
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++                                   1473
BEING MARKER1 SINCE MARKER2 BY A MARKER3 FCR THE SUBJECT CF MARKER4 QUEEN CF SC 1474
OTS, I WISHED TC TEST FCR MYSELF THE TRUTH MARKER5 CF THE MANY LEGENDS WHICH SU 1475
RROUNDED HER NAME.                                                              1476
BEING &1 SINCE &2 BY A &3 FCR THE SUBJECT CF &4 QUEEN OF SCOTS, I WISHED TC TES 1477
T FOR MYSELF THE TRUTH &5 CF THE MANY LEGENDS WHICH SURRCUNDED HER NAME.        1478
BEING POSSESSED SINCE CHILDHCOD BY A PASSICN FOR THE SUBJECT OF MARY QUEEN OF S 1479
COTS, I WISHED TC TEST FCR MYSELF THE TRUTH CR FALSEHCOD CF THE MANY LEGENDS WH 1480
ICH SURRCUNDED HER NAME.                                                        1481
FIND OUT WHAT MARKER1 MUST HAVE BEEN LIKE A PERSCN.                             1482
```

```
FIND OUT WHAT &1 MUST HAVE BEEN LIKE A PERSON.                                  1483
FIND OUT WHAT MARY QUEEN OF SCOTS MUST HAVE BEEN LIKE A PERSON.                 1484
                                                                                1485
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                   1486
                                                                                1487
                                                                                1488
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                   1489
--------------------------------------------------------------------------      1490
**********TEST-5 PF FUNCTION**************************************             1491
+++++THIS PS FUNCTION HAS NULL STRING+++++                                       1492
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                   1493
THESE STARK CONDITIONS FOUND A BLEAK PARALLEL IN THE POLITICAL CLIMATE WHICH    1494
 THEN PREVAILED BETWEEN COUNTRIES.                                              1495
THESE ARRANGEMENTS, LIKE THE STEPS OF A FORMAL DANCE, CONVEY LITTLE OF THE FEEL 1496
INGS OF THE PEOPLE CONCERNED.                                                   1497
THE&1ITIONS FOUND A &2 IN THE POLITIC&3ICH    THEN PREVAILED BETWEEN COUNTRIES. 1498
&2 ARRANGEMENTS, LIKE THE STEPS &4 A &1, CONVEY LITTLE &4 THE FEELINGS &4 &3 CO 1499
NCERNED.                                                                        1500
                                                                                1501
                                                                                1502
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                   1503
THESE STARK CONDITIONS FOUND A BLEAK PARALLEL IN THE POLITICAL CLIMATE WHICH    1504
 THEN PREVAILED BETWEEN COUNTRIES.                                              1505
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                   1506
THESE ARRANGEMENTS, LIKE THE STEPS OF A FORMAL DANCE, CONVEY LITTLE OF THE FEEL 1507
INGS OF THE PEOPLE CONCERNED.                                                   1508
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                   1509
                                                                                1510
                                                                                1511
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                   1512
--------------------------------------------------------------------------      1513
**********TEST-6 PF FUNCTION**************************************             1514
+++++THIS PS FUNCTION HAS NULL STRING+++++                                       1515
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                   1516
SHE HAD BEEN MARRIED AT THE AGE OF NINETEEN.                                    1517
&1 &2 AT THE AGE OF &3.                                                         1518
&(PS,&(CL,&(RS)))                                                               1519
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               1520
```

```
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1521
HE DIED AT THE AGE OF 55.                                                       1522
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               1523
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1524
HE WAS MARRIED AT THE AGE OF 76.                                                1525
                                                                                1526
                                                                                1527
                                                                                1528
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                   1529
                                                                                1530
                                                                                1531
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                   1532
--------------------------------------------------------------------------------1533
**********TEST-7 PF FUNCTION**************************************************1534
+++++THIS PS FUNCTION HAS NULL STRING+++++                                       1535
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                   1536
THE MARRIAGE OF JAMES AND MARY DOES NOT SEEM TO HAVE BEEN A PARTICULARLY HAPPY  1537
ONE IN ITS EARLY STAGES.                                                        1538
THE MARRIAGE OF &1 DOES NOT SEEM TO HAVE BEEN A PARTICULARLY &2 ONE IN ITS EARL 1539
Y STAGES.                                                                       1540
&(CL,A,JOHN AND MARGARET,UNHAPPY)                                               1541
&(CL,B)                                                                         1542
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               1543
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1544
&(CL,A,JOHN AND MARGARET,UNHAPPY)                                               1545
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               1546
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1547
THE MARRIAGE OF JOHN AND MARGARET DOES NOT SEEM TO HAVE BEEN A PARTICULARLY UNH 1548
APPY ONE IN ITS EARLY STAGES.                                                   1549
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                   1550
                                                                                1551
                                                                                1552
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                   1553
--------------------------------------------------------------------------------1554
**********TEST-8 PF FUNCTION**************************************************1555
+++++THIS PS FUNCTION HAS NULL STRING+++++                                       1556
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                   1557
        **FORMS  IN  STORE**                                                    1558
```

```
+++++THIS PS FUNCTION HAS NULL STRING+++++                                1559
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++                             1560
************************************************************************ 1561
+++++THIS PS FUNCTION HAS NULL STRING+++++                                1562
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++                             1563
**THE DEFINITION OF FORMS BEGINS**                                        1564
THE ONLY THING WHICH WAS NOT TAKEN FROM                                   1565
MARY WAS HER ACTUAL MONEY, ON WHICH                                       1566
 SHE DEPENDED FOR PAYING HER SERVANTS AND                                 1567
FOR HER OWN NECESSITIES.THIS SHE FOUND                                    1568
+++++ THERE IS NOT SUCH A FORM IN STORE+++++                              1569
+++++PF"X5                                                               1570
STILL IN THE CUPBOARD WHERE SHE HAD LEFT IT                               1571
BUT LATER DIRECTIVES CAME FROM LONDON THAT THIS TOO WAS TO BE SEIZED.     1572
&2E &4LY &2ING&1HICH&1AS &5T &3 FROM                                      1573
&4&2RY W&2S HER &2CTU&2&3 &4C&5, ON WH&6&1                                1574
 SHE &3 FOR &2 HER &1 AND                                                 1575
F&1 &2R OWN &3.THIS S&2 &4ND                                              1576
STILL IN TH&5 &3UP&2O&1R&4 WH&5R&5 SH&5 H&1&4 L&5&&T IT                   1577
&2 &3 DIRECTIVES CAME FROM LONDON THAT THIS TOO WAS TO BE &1.             1578
THE ONLY THING WHICH WAS NOT TAKEN FROM MARY WAS HER ACTUAL MONEY, ON WHICH SHE 1579
 DEPENDED FOR PAYING HER SERVANTS AND FOR HER OWN NECESSITIES.THIS SHE FOUND ST 1580
ILL IN THE CUPBOARD WHERE SHE HAD LEFT IT BUT LATER DIRECTIVES CAME FROM LONDON 1581
 THAT THIS TOO WAS TO BE SEIZED.                                          1582
+++++THIS PS FUNCTION HAS NULL STRING+++++                                1583
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++                             1584
                                                                          1585
                                                                          1586
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++                             1587
------------------------------------------------------------------------ 1588
**********TEST-1 EQ FUNCTION**************************************** 1589
NAME  LINE1                                                               1590
NAME  LINE2                                                               1591
NAME  EQUAL                                                               1592
+++++THERE ARE NO FORMS IN STORE+++++                                     1593
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                         1594
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++          1595
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                         1596
```

```
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
SPRING,THE SWEET SPRING,IS THE YEAR   S PLEASANT KING
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
THEN BLOOMS EACH THING,THEN MAIDS DANCE IN A RING.
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
THEN BLOOMS EACH THING,THEN MAIDS DANCE IN A RING.
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
SPRING,THE SWEET SPRING,IS THE YEAR   S PLEASANT KING
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
SPRING,THE SWEET SPRING,IS THE YEAR   S PLEASANT KING
THEN BLOOMS EACH THING,THEN MAIDS DANCE IN A RING.
£(EQ,£(RS),LINE1,(£(PS,££(CL,LINE1))),(£(PS,££(CL,LINE2))))£(CL,EQUAL)
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
THEN BLOOMS EACH THING,THEN MAIDS DANCE IN A RING.
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++


+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++
--------------------------------------------------------------------------
**********TEST-1 EQ FUNCTION**WITH TRACE**********
PS"
+++++THIS PS FUNCTION HAS NULL STRING+++++
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++
RS
RS
RS
```

```
DS"LINE1"SPRING,THE SWEET SPRING,IS THE YEAR   S PLEASANT KING                                                    1635
RS                                                                                                               1636
RS                                                                                                               1637
DS"LINE2"THEN BLCCMS EACH THING,THEN MAIDS CANCE IN A RING.                                                      1638
DS"EQUAL"£(EQ,£(RS),LINE1,(£(PS,££(CL,LINE1))),(£(PS,££(CL,LINE2))))£(CL,EQUAL)                                  1639
LN"NAME                                                                                                          1640
NAME  LINE1                                                                                                      1641
NAME  LINE2                                                                                                      1642
NAME  EQUAL                                                                                                      1643
+++++THERE ARE NC FCRMS IN STCRE+++++                                                                            1644
CL"EQUAL                                                                                                         1645
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                                                                1646
+++++FCRM WITH NO MARKERS ,ARGUMENTS CF CL FUNCTICN IGNORED+++++                                                  1647
RS                                                                                                               1648
EQ"LINE1"LINE1"£(PS,££(CL,LINE1))"£(PS,££(CL,LINE2))                                                            1649
CL"LINE1                                                                                                         1650
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                                                                1651
+++++FCRM WITH NO MARKERS ,ARGUMENTS CF CL FUNCTICN IGNORED+++++                                                  1652
PS"SPRING,THE SWEET SPRING,IS THE YEAR   S PLEASANT KING                                                         1653
SPRING,THE SWEET SPRING,IS THE YEAR   S PLEASANT KING                                                            1654
CL"EQUAL                                                                                                         1655
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                                                                1656
+++++FCRM WITH NO MARKERS ,ARGUMENTS CF CL FUNCTICN IGNORED+++++                                                  1657
RS                                                                                                               1658
EQ"LINE2"LINE1"£(PS,££(CL,LINE1))"£(PS,££(CL,LINE2))                                                            1659
CL"LINE2                                                                                                         1660
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                                                                1661
+++++FCRM WITH NO MARKERS ,ARGUMENTS CF CL FUNCTICN IGNORED+++++                                                  1662
PS"THEN BLCCMS EACH THING,THEN MAIDS CANCE IN A RING.                                                            1663
THEN BLCCMS EACH THING,THEN MAIDS CANCE IN A RING.                                                               1664
CL"EQUAL                                                                                                         1665
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                                                                1666
+++++FCRM WITH NO MARKERS ,ARGUMENTS CF CL FUNCTICN IGNORED+++++                                                  1667
RS                                                                                                               1668
EQ"ANY NAME"LINE1"£(PS,££(CL,LINE1))"£(PS,££(CL,LINE2))                                                         1669
CL"LINE2                                                                                                         1670
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                                                                1671
+++++FCRM WITH NO MARKERS ,ARGUMENTS CF CL FUNCTICN IGNORED+++++                                                  1672
```

```
PS"THEN BLOOMS EACH THING,THEN MAIDS DANCE IN A RING.                    1673
THEN BLOOMS EACH THING,THEN MAIDS DANCE IN A RING.                       1674
CL"EQUAL                                                                 1675
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                        1676
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++         1677
RS                                                                       1678
EQ"LINE1"LINE1"£(PS,££(CL,LINE1))"£(PS,££(CL,LINE2))                    1679
CL"LINE1                                                                 1680
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                        1681
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++         1682
PS"SPRING,THE SWEET SPRING,IS THE YEAR   S PLEASANT KING                 1683
SPRING,THE SWEET SPRING,IS THE YEAR   S PLEASANT KING                    1684
CL"EQUAL                                                                 1685
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                        1686
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++         1687
RS                                                                       1688
PF"LINE1                                                                 1689
SPRING,THE SWEET SPRING,IS THE YEAR   S PLEASANT KING                    1690
PF"LINE2                                                                 1691
THEN BLOOMS EACH THING,THEN MAIDS DANCE IN A RING.                       1692
PF"EQUAL                                                                 1693
£(EQ,£(RS),LINE1,(£(PS,££(CL,LINE1))),(£(PS,££(CL,LINE2))))£(CL,EQUAL)  1694
EQ""LINE1"£(PS,££(CL,LINE1))"£(PS,££(CL,LINE2))                          1695
CL"LINE2                                                                 1696
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                        1697
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++         1698
PS"THEN BLOOMS EACH THING,THEN MAIDS DANCE IN A RING.                    1699
THEN BLOOMS EACH THING,THEN MAIDS DANCE IN A RING.                       1700
CL"EQUAL                                                                 1701
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                        1702
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++         1703
RS                                                                       1704
                                                                         1705
                                                                         1706
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                            1707
------------------------------------------------------------------------ 1708
**********TEST-2 EQ FUNCTION**************************************** 1709
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                        1710
```

```
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
TEXT**********NOT ALL THAT 1 YOUR 2 EYES
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
TEXT**********NOT ALL THAT MM1 YOUR MM2 EYES
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
TEXT**********NOT ALL THAT MARKER1 YOUR MARKER2 EYES
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
TEXT**********NOT ALL THAT SLEEPING YOUR PLEASING EYES
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
```

```
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1749
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               1750
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1751
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               1752
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1753
END  OF PROGRAM                                                                 1754
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               1755
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1756
TEMP**********NOT ALL THAT TEMPTS YOUR WANDERING EYES                          1757
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               1758
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1759
TEXT**********NOT ALL THAT TEMPTS YOUR WANDERING EYES                          1760
+++++THIS PS FUNCTION HAS NULL STRING+++++                                      1761
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                   1762
--------------------------------------------------------------------------      1763
+++++THIS PS FUNCTION HAS NULL STRING+++++                                      1764
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                   1765
                                                                                1766
                                                                                1767
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                   1768
--------------------------------------------------------------------------      1769
********************TEST-2 EQ FUNCTION**WITH  TRACE************************    1770
PS"                                                                             1771
+++++THIS PS FUNCTION HAS NULL STRING+++++                                      1772
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                   1773
RS                                                                              1774
RS                                                                              1775
DS"TEXT"NOT ALL THAT TEMPTS YOUR WANDERING EYES                                 1776
CL"TEXT                                                                         1777
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               1778
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1779
DS"TEMP"NOT ALL THAT TEMPTS YOUR WANDERING EYES                                 1780
RS                                                                              1781
SS"TEXT"TEMPTS"WANDERING                                                        1782
DS"PROC"£(DS,TEXT,££(CL,TEXT,£(RS)))£(EQ,££(CL,TEMP),££(CL,TEXT),(£(PS,END  OF 1783
PROGRAM)£(PS,TEMP**********££(CL,TEMP))£(PS,TEXT**********££(CL,TEXT))),(£(PS,T 1784
EXT**********££(CL,TEXT))£(SS,TEXT,£(RS))£(CL,PROC)))                          1785
CL"PROC                                                                         1786
```

```
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                1787
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1788
RS                                                                              1789
CL"TEXT"1"2                                                                     1790
DS"TEXT"NOT ALL THAT 1 YOUR 2 EYES                                              1791
CL"TEMP                                                                         1792
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                1793
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1794
CL"TEXT                                                                         1795
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                1796
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1797
EQ"NOT ALL THAT TEMPTS YOUR WANDERING EYES"NOT ALL THAT 1 YOUR 2 EYES"£(PS,END  1798
 OF PROGRAM)£(PS,TEMP**********££(CL,TEMP))£(PS,TEXT**********££(CL,TEXT))"£(PS 1799
,TEXT**********££(CL,TEXT))£(SS,TEXT,£(RS))£(CL,PROC)                          1800
CL"TEXT                                                                         1801
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                1802
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1803
PS"TEXT**********NOT ALL THAT 1 YOUR 2 EYES                                     1804
TEXT**********NOT ALL THAT 1 YOUR 2 EYES                                        1805
RS                                                                              1806
SS"TEXT"1"2                                                                     1807
CL"PROC                                                                         1808
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                1809
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1810
RS                                                                              1811
CL"TEXT"MM1"MM2                                                                 1812
DS"TEXT"NOT ALL THAT MM1 YOUR MM2 EYES                                          1813
CL"TEMP                                                                         1814
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                1815
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1816
CL"TEXT                                                                         1817
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                1818
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1819
EQ"NOT ALL THAT TEMPTS YOUR WANDERING EYES"NOT ALL THAT MM1 YOUR MM2 EYES"£(PS,  1820
END  OF PROGRAM)£(PS,TEMP**********££(CL,TEMP))£(PS,TEXT**********££(CL,TEXT))"  1821
£(PS,TEXT**********££(CL,TEXT))£(SS,TEXT,£(RS))£(CL,PROC)                      1822
CL"TEXT                                                                         1823
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                1824
```

```
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1825
PS"TEXT**********NOT ALL THAT MM1 YOUR MM2 EYES                                 1826
TEXT**********NOT ALL THAT MM1 YOUR MM2 EYES                                    1827
RS                                                                              1828
SS"TEXT"MM1"MM2                                                                 1829
CL"PROC                                                                         1830
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               1831
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1832
RS                                                                              1833
CL"TEXT"MARKER1"MARKER2                                                         1834
DS"TEXT"NOT ALL THAT MARKER1 YOUR MARKER2 EYES                                  1835
CL"TEMP                                                                         1836
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               1837
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1838
CL"TEXT                                                                         1839
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               1840
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1841
EQ"NOT ALL THAT TEMPTS YOUR WANDERING EYES"NOT ALL THAT MARKER1 YOUR MARKER2 EY 1842
ES"£(PS,END  OF PROGRAM)£(PS,TEMP**********££(CL,TEMP))£(PS,TEXT**********££(CL 1843
,TEXT))"£(PS,TEXT**********££(CL,TEXT))£(SS,TEXT,£(RS))£(CL,PROC)              1844
CL"TEXT                                                                         1845
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               1846
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1847
PS"TEXT**********NOT ALL THAT MARKER1 YOUR MARKER2 EYES                         1848
TEXT**********NOT ALL THAT MARKER1 YOUR MARKER2 EYES                            1849
RS                                                                              1850
SS"TEXT"MARKER1"MARKER2                                                         1851
CL"PROC                                                                         1852
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               1853
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1854
RS                                                                              1855
CL"TEXT"SLEEPING"PLEASING                                                       1856
DS"TEXT"NOT ALL THAT SLEEPING YOUR PLEASING EYES                                1857
CL"TEMP                                                                         1858
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               1859
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1860
CL"TEXT                                                                         1861
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               1862
```

```
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                  1863
EQ"NOT ALL THAT TEMPTS YOUR WANDERING EYES"NOT ALL THAT SLEEPING YOUR PLEASING  1864
EYES"£(PS,END  OF PROGRAM)£(PS,TEMP**********££(CL,TEMP))£(PS,TEXT**********££( 1865
CL,TEXT))"£(PS,TEXT**********££(CL,TEXT))£(SS,TEXT,£(RS))£(CL,PROC)            1866
CL"TEXT                                                                         1867
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               1868
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1869
PS"TEXT**********NOT ALL THAT SLEEPING YOUR PLEASING EYES                       1870
TEXT**********NOT ALL THAT SLEEPING YOUR PLEASING EYES                         1871
RS                                                                              1872
SS"TEXT"SLEEPING"PLEASING                                                       1873
CL"PROC                                                                         1874
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               1875
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1876
RS                                                                              1877
CL"TEXT"TEMPTS"WANDERING                                                        1878
DS"TEXT"NOT ALL THAT TEMPTS YOUR WANDERING EYES                                 1879
CL"TEMP                                                                         1880
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               1881
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1882
CL"TEXT                                                                         1883
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               1884
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1885
EQ"NOT ALL THAT TEMPTS YOUR WANDERING EYES"NOT ALL THAT TEMPTS YOUR WANDERING E 1886
YES"£(PS,END  OF PROGRAM)£(PS,TEMP**********££(CL,TEMP))£(PS,TEXT**********££(C 1887
L,TEXT))"£(PS,TEXT**********££(CL,TEXT))£(SS,TEXT,£(RS))£(CL,PROC)            1888
PS"END  OF PROGRAM                                                              1889
END  OF PROGRAM                                                                 1890
CL"TEMP                                                                         1891
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               1892
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1893
PS"TEMP**********NOT ALL THAT TEMPTS YOUR WANDERING EYES                        1894
TEMP**********NOT ALL THAT TEMPTS YOUR WANDERING EYES                          1895
CL"TEXT                                                                         1896
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               1897
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1898
PS"TEXT**********NOT ALL THAT TEMPTS YOUR WANDERING EYES                        1899
TEXT**********NOT ALL THAT TEMPTS YOUR WANDERING EYES                          1900
```

```
PS"                                                                          1901
+++++THIS PS FUNCTION HAS NULL STRING+++++                                   1902
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                1903
RS                                                                           1904
PS"-------------------------------------------------------------------------  1905
---------------------------------------------------------------------------  1906
PS"                                                                          1907
+++++THIS PS FUNCTION HAS NULL STRING+++++                                   1908
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                1909
RS                                                                           1910
                                                                             1911
                                                                             1912
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                1913
**********TEST-3 EQ FUNCTION*****************************************       1914
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                            1915
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++             1916
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                            1917
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++             1918
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                            1919
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++             1920
AND I1 ILKA BUSH J AROUND L1 HINGING                                         1921
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                            1922
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++             1923
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                            1924
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++             1925
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                            1926
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++             1927
AND II ILKA BUSH JJ AROUND LL HINGING                                        1928
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                            1929
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++             1930
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                            1931
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++             1932
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                            1933
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++             1934
AND III ILKA BUSH JJJ AROUND LLL HINGING                                     1935
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                            1936
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++             1937
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                            1938
```

```
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1939
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               1940
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1941
AND IIII ILKA BUSH JJJJ AROUND LLLL HINGING                                     1942
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               1943
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1944
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               1945
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1946
DOWN IN YON GARDEN SWEET AND GAY WHERE BONNIE GROWS THE LILY                    1947
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               1948
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1949
AND SEE THE BIRDS ON ILKA BUSH AND LEAVES AROUND THEM HINGING                   1950
 PS,------------------------------------------------------------------------   1951
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                   1952
-------------------------------------------------------------------------       1953
+++++THIS PS FUNCTION HAS NULL STRING+++++                                       1954
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                   1955
                                                                                1956
                                                                                1957
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                   1958
-------------------------------------------------------------------------       1959
**********TEST-3 EQ FUNCTION***WITH  TRACE**********                           1960
PS"                                                                             1961
+++++THIS PS FUNCTION HAS NULL STRING+++++                                       1962
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                   1963
RS                                                                              1964
RS                                                                              1965
DS"V1"AND II ILKA BUSH JJ AROUND LL HINGING                                     1966
RS                                                                              1967
DS"V2"AND SEE THE BIRDS ON ILKA BUSH AND LEAVES AROUND THEM HINGING             1968
RS                                                                              1969
DS"V3"DOWN 1 SWEET 2 BONNIE 3 LILY                                             1970
SS"V3"1"2"3                                                                     1971
DS"EQUAL"$(EQ,$(SS,V1,$(RS))$$(CL,V1,$(RS)),$$(CL,V2),($(PS,$$(CL,V3,$(RS)))$(P 1972
S,$$(CL,V2))),($(DS,V1,$$(CL,V1,$(RS)))$(PS,$$(CL,V1))$(CL,EQUAL))              1973
CL"EQUAL                                                                        1974
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               1975
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1976
```

```
RS                                                                              1977
SS"V1"I1"JJ"LL                                                                  1978
RS                                                                              1979
CL"V1"I1"J"L1                                                                   1980
CL"V2                                                                           1981
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               1982
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1983
EQ"AND I1 ILKA BUSH J AROUND L1 HINGING"AND SEE THE BIRDS ON ILKA BUSH AND LEAV 1984
ES AROUND THEM HINGING"£(PS,££(CL,V3,£(RS)))£(PS,££(CL,V2))"£(DS,V1,££(CL,V1,£( 1985
RS)))£(PS,££(CL,V1))£(CL,EQUAL)                                                 1986
RS                                                                              1987
CL"V1"I1"J"L1                                                                   1988
DS"V1"AND I1 ILKA BUSH J AROUND L1 HINGING                                      1989
CL"V1                                                                           1990
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               1991
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1992
PS"AND I1 ILKA BUSH J AROUND L1 HINGING                                         1993
AND I1 ILKA BUSH J AROUND L1 HINGING                                            1994
CL"EQUAL                                                                        1995
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               1996
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 1997
RS                                                                              1998
SS"V1"I1"J"L1                                                                   1999
RS                                                                              2000
CL"V1"II"JJ"LL                                                                  2001
CL"V2                                                                           2002
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               2003
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2004
EQ"AND II ILKA BUSH JJ AROUND LL HINGING"AND SEE THE BIRDS ON ILKA BUSH AND LEA 2005
VES AROUND THEM HINGING"£(PS,££(CL,V3,£(RS)))£(PS,££(CL,V2))"£(DS,V1,££(CL,V1,£ 2006
(RS)))£(PS,££(CL,V1))£(CL,EQUAL)                                                2007
RS                                                                              2008
CL"V1"II"JJ"LL                                                                  2009
DS"V1"AND II ILKA BUSH JJ AROUND LL HINGING                                     2010
CL"V1                                                                           2011
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               2012
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2013
PS"AND II ILKA BUSH JJ AROUND LL HINGING                                        2014
```

```
AND II ILKA BUSH JJ ARCUND LL HINGING                                    2015
CL"EQUAL                                                                 2016
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                        2017
+++++FCRM WITH NC MARKERS ,ARGUMENTS OF CL FUNCTICN IGNORED+++++         2018
RS                                                                       2019
SS"V1"II"JJ"LL                                                           2020
RS                                                                       2021
CL"V1"III"JJJ"LLL                                                        2022
CL"V2                                                                    2023
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                        2024
+++++FCRM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTICN IGNORED+++++         2025
EQ"AND III ILKA BUSH JJJ ARCUND LLL HINGING"AND SEE THE BIRDS ON ILKA BUSH AND  2026
LEAVES ARCUND THEM HINGING"£(PS,££(CL,V3,£(RS)))£(PS,££(CL,V2))"£(DS,V1,££(CL,V 2027
1,£(RS)))£(PS,££(CL,V1))£(CL,EQUAL)                                     2028
RS                                                                       2029
CL"V1"III"JJJ"LLL                                                        2030
DS"V1"AND III ILKA BUSH JJJ ARCUND LLL HINGING                           2031
CL"V1                                                                    2032
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                        2033
+++++FCRM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTICN IGNORED+++++         2034
PS"AND III ILKA BUSH JJJ ARCUND LLL HINGING                              2035
AND III ILKA BUSH JJJ ARCUND LLL HINGING                                 2036
CL"EQUAL                                                                 2037
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                        2038
+++++FCRM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTICN IGNORED+++++         2039
RS                                                                       2040
SS"V1"III"JJJ"LLL                                                        2041
RS                                                                       2042
CL"V1"IIII"JJJJ"LLLL                                                     2043
CL"V2                                                                    2044
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++                        2045
+++++FCRM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTICN IGNORED+++++         2046
EQ"AND IIII ILKA BUSH JJJJ ARCUND LLLL HINGING"AND SEE THE BIRDS ON ILKA BUSH A 2047
ND LEAVES ARCUND THEM HINGING"£(PS,££(CL,V3,£(RS)))£(PS,££(CL,V2))"£(DS,V1,££(C 2048
L,V1,£(RS)))£(PS,££(CL,V1))£(CL,EQUAL)                                  2049
RS                                                                       2050
CL"V1"IIII"JJJJ"LLLL                                                     2051
DS"V1"AND IIII ILKA BUSH JJJJ ARCUND LLLL HINGING                        2052
```

```
CL"V1                                                                    2053
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                        2054
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++         2055
PS"AND IIII ILKA BUSH JJJJ AROUND LLLL HINGING                           2056
AND IIII ILKA BUSH JJJJ AROUND LLLL HINGING                              2057
CL"EQUAL                                                                 2058
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                        2059
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++         2060
RS                                                                       2061
SS"V1"IIII"JJJJ"LLLL                                                     2062
RS                                                                       2063
CL"V1"SEE THE BIRDS ON"AND LEAVES"THEM                                   2064
CL"V2                                                                    2065
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                        2066
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++         2067
EQ"AND SEE THE BIRDS ON ILKA BUSH AND LEAVES AROUND THEM HINGING"AND SEE THE BI 2068
RDS ON ILKA BUSH AND LEAVES AROUND THEM HINGING"£(PS,££(CL,V3,£(RS)))£(PS,££(CL 2069
,V2))"£(DS,V1,££(CL,V1,£(RS)))£(PS,££(CL,V1))£(CL,EQUAL)                 2070
RS                                                                       2071
CL"V3"IN YON GARDEN"AND GAY WHERE"GROWS THE                              2072
PS"DOWN IN YON GARDEN SWEET AND GAY WHERE BONNIE GROWS THE LILY          2073
DOWN IN YON GARDEN SWEET AND GAY WHERE BONNIE GROWS THE LILY             2074
CL"V2                                                                    2075
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                        2076
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++         2077
PS"AND SEE THE BIRDS ON ILKA BUSH AND LEAVES AROUND THEM HINGING         2078
AND SEE THE BIRDS ON ILKA BUSH AND LEAVES AROUND THEM HINGING            2079
PS"                                                                      2080
+++++THIS PS FUNCTION HAS NULL STRING+++++                               2081
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                            2082
RS                                                                       2083
PS"-------------------------------------------------------------------- 2084
----------------------------------------------------------------------   2085
PS"                                                                      2086
+++++THIS PS FUNCTION HAS NULL STRING+++++                               2087
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                            2088
RS                                                                       2089
                                                                         2090
```

```
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++
--------------------------------------------------------------------------------
**********TEST-4 EQ FUNCTION**********
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
WHEN MU1 MA2 YOU3 IN EARLY GREECE SHE SUNG
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
QQQQ*****£(PS,WHEN MU1 MA2 YOU3 IN EARLY GREECE SHE SUNG)£(PS,QQQQ*****££(CL,QQ
QQ))£(SS,QQQQ,£(RS))£(DS,QQQQ,££(CL,QQQQ,£(RS)))£(EQ,££(CL,COUNTER),££(CL,UNIT)
,(£(PS,END PROGRAM)£(PS,QQQQ*******££(CL,QQQQ))£(PS,COUNTER****££(CL,COUNTER))£
(PS,UNIT*******££(CL,UNIT))),(£(DS,UNIT,I££(CL,UNIT))£(PS,UNIT*****££(CL,UNIT))
£(CL,QQQQ)))
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
UNIT*****II
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
WHEN MUM1 MAM2 YOUM3 IN EARLY GREECE SHE SUNG
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
QQQQ*****£(PS,WHEN MUM1 MAM2 YOUM3 IN EARLY GREECE SHE SUNG)£(PS,QQQQ*****££(CL
,QQQQ))£(SS,QQQQ,£(RS))£(DS,QQQQ,££(CL,QQQQ,£(RS)))£(EQ,££(CL,COUNTER),££(CL,UN
IT),(£(PS,END PROGRAM)£(PS,QQQQ*******££(CL,QQQQ))£(PS,COUNTER****££(CL,COUNTER
))£(PS,UNIT*******££(CL,UNIT))),(£(DS,UNIT,I££(CL,UNIT))£(PS,UNIT*****££(CL,UNI
T))£(CL,QQQQ)))
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
```

```
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2129
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               2130
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2131
UNIT*****III                                                                    2132
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               2133
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2134
WHEN MUMM1 MAMM2 YOUMM3 IN EARLY GREECE SHE SUNG                                2135
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               2136
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2137
QQQQ******£(PS,WHEN MUMM1 MAMM2 YOUMM3 IN EARLY GREECE SHE SUNG)£(PS,QQQQ******££ 2138
(CL,QQQQ))£(SS,QQQQ,£(RS))£(DS,QQQQ,££(CL,QQQQ,£(RS)))£(EQ,££(CL,COUNTER),££(CL 2139
,UNIT),(£(PS,END PROGRAM)£(PS,QQQQ*******££(CL,QQQQ))£(PS,COUNTER*****££(CL,COUN 2140
TER))£(PS,UNIT*******££(CL,UNIT))),(£(DS,UNIT,I££(CL,UNIT))£(PS,UNIT******££(CL, 2141
UNIT))£(CL,QQQQ)))                                                              2142
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               2143
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2144
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               2145
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2146
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               2147
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2148
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               2149
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2150
UNIT*****IIII                                                                   2151
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               2152
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2153
WHEN MUMMM1 MAMMM2 YOUMMM3 IN EARLY GREECE SHE SUNG                             2154
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               2155
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2156
QQQQ******£(PS,WHEN MUMMM1 MAMMM2 YOUMMM3 IN EARLY GREECE SHE SUNG)£(PS,QQQQ**** 2157
*££(CL,QQQQ))£(SS,QQQQ,£(RS))£(DS,QQQQ,££(CL,QQQQ,£(RS)))£(EQ,££(CL,COUNTER),££ 2158
(CL,UNIT),(£(PS,END PROGRAM)£(PS,QQQQ*******££(CL,QQQQ))£(PS,COUNTER*****££(CL,C 2159
OUNTER))£(PS,UNIT*******££(CL,UNIT))),(£(DS,UNIT,I££(CL,UNIT))£(PS,UNIT******££( 2160
CL,UNIT))£(CL,QQQQ)))                                                           2161
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               2162
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2163
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               2164
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2165
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               2166
```

```
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2167
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               2168
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2169
UNIT*****IIIII                                                                  2170
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               2171
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2172
WHEN MUMMM1 MAMMM2 YOUMMM3 IN EARLY GREECE SHE SUNG                             2173
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               2174
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2175
QQQQ******£(PS,WHEN MUMMM1 MAMMM2 YOUMMM3 IN EARLY GREECE SHE SUNG)£(PS,QQQQ* 2176
****££(CL,QQQQ))£(SS,QQQQ,£(RS))£(DS,QQQQ,££(CL,QQQQ,£(RS)))£(EQ,££(CL,COUNTER) 2177
,££(CL,UNIT),(£(PS,END PROGRAM)£(PS,QQQQ*******££(CL,QQQQ))£(PS,COUNTER****££(C 2178
L,COUNTER))£(PS,UNIT*******££(CL,UNIT))),(£(DS,UNIT,I££(CL,UNIT))£(PS,UNIT***** 2179
££(CL,UNIT))£(CL,QQQQ)))                                                        2180
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               2181
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2182
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               2183
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2184
END PROGRAM                                                                     2185
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               2186
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2187
QQQQ********£(PS,WHEN MUSIC,HEAVENLY MAID,WAS YOUNG,WHILE YET IN EARLY GREECE SH 2188
E SUNG)£(PS,QQQQ*******££(CL,QQQQ))£(SS,QQQQ,£(RS))£(DS,QQQQ,££(CL,QQQQ,£(RS)))£( 2189
EQ,££(CL,COUNTER),££(CL,UNIT),(£(PS,END PROGRAM)£(PS,QQQQ*******££(CL,QQQQ))£(P 2190
S,COUNTER****££(CL,COUNTER))£(PS,UNIT*******££(CL,UNIT))),(£(DS,UNIT,I££(CL,UNI 2191
T))£(PS,UNIT******££(CL,UNIT))£(CL,QQQQ)))                                      2192
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               2193
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2194
COUNTER****IIIII                                                                2195
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               2196
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2197
UNIT*******IIIII                                                                2198
+++++THIS PS FUNCTION HAS NULL STRING+++++                                       2199
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                    2200
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               2201
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2202
WHEN MUSIC"HEAVENLY MAID"WAS YOUNG"WHILE YET IN EARLY GREECE SHE SUNG           2203
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               2204
```

```
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
QQQQ******£(PS,WHEN MUSIC,HEAVENLY MAID,WAS YOUNG,WHILE YET IN EARLY GREECE SHE
SUNG)£(PS,QQQQ******££(CL,QQQQ))£(SS,QQQQ,£(RS))£(DS,QQQQ,££(CL,QQQQ,£(RS)))£(EQ
,££(CL,COUNTER),££(CL,UNIT),(£(PS,END PROGRAM)£(PS,QQQQ********££(CL,QQQQ))£(PS,
COUNTER*****££(CL,COUNTER))£(PS,UNIT********££(CL,UNIT))),(£(DS,UNIT,I££(CL,UNIT)
)£(PS,UNIT******££(CL,UNIT))£(CL,QQQQ)))
--------------------------------------------------------------------------
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++SS"QQQQ"


+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++
--------------------------------------------------------------------------
********************TEST-4 EQ FUNCTION**WITH  TRACE************************
PS"
+++++THIS PS FUNCTION HAS NULL STRING+++++
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++
RS
DS"COUNTER"IIIII
DS"UNIT"I
DS"QQQQ"£(PS,WHEN MU1 MA2 YOU3 IN EARLY GREECE SHE SUNG)£(PS,QQQQ******££(CL,QQQ
Q))£(SS,QQQQ,£(RS))£(DS,QQQQ,££(CL,QQQQ,£(RS)))£(EQ,££(CL,COUNTER),££(CL,UNIT),
(£(PS,END PROGRAM)£(PS,QQQQ********££(CL,QQQQ))£(PS,COUNTER*****££(CL,COUNTER))£(
PS,UNIT********££(CL,UNIT))),(£(DS,UNIT,I££(CL,UNIT))£(PS,UNIT******££(CL,UNIT))£
(CL,QQQQ)))
CL"QQQQ
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
PS"WHEN MU1 MA2 YOU3 IN EARLY GREECE SHE SUNG
WHEN MU1 MA2 YOU3 IN EARLY GREECE SHE SUNG
CL"QQQQ
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
PS"QQQQ******£(PS,WHEN MU1 MA2 YOU3 IN EARLY GREECE SHE SUNG)£(PS,QQQQ******££(CL
,QQQQ))£(SS,QQQQ,£(RS))£(DS,QQQQ,££(CL,QQQQ,£(RS)))£(EQ,££(CL,COUNTER),££(CL,UN
IT),(£(PS,END PROGRAM)£(PS,QQQQ********££(CL,QQQQ))£(PS,COUNTER*****££(CL,COUNTER
))£(PS,UNIT********££(CL,UNIT))),(£(DS,UNIT,I££(CL,UNIT))£(PS,UNIT******££(CL,UNI
T))£(CL,QQQQ)))
```

```
QQQQ*****£(PS,WHEN MU1 MA2 YOU3 IN EARLY GREECE SHE SUNG)£(PS,QQQQ*****££(CL,QQ
QQ))£(SS,QQQQ,£(RS))£(DS,QQQQ,££(CL,QQQQ,£(RS)))£(EQ,££(CL,COUNTER),££(CL,UNIT)
,(£(PS,END PROGRAM)£(PS,QQQQ*******££(CL,QQQQ))£(PS,COUNTER****££(CL,COUNTER))£
(PS,UNIT*******££(CL,UNIT))),(£(DS,UNIT,I££(CL,UNIT))£(PS,UNIT*****££(CL,UNIT))
£(CL,QQQQ)))
RS
SS"QQQQ"1"2"3
RS
CL"QQQQ"M1"M2"M3
DS"QQQQ"£(PS,WHEN MUM1 MAM2 YOUM3 IN EARLY GREECE SHE SUNG)£(PS,QQQQ*****££(CL,
QQQQ))£(SS,QQQQ,£(RS))£(DS,QQQQ,££(CL,QQQQ,£(RS)))£(EQ,££(CL,COUNTER),££(CL,UNI
T),(£(PS,END PROGRAM)£(PS,QQQQ*******££(CL,QQQQ))£(PS,COUNTER****££(CL,COUNTER)
)£(PS,UNIT*******££(CL,UNIT))),(£(DS,UNIT,I££(CL,UNIT))£(PS,UNIT*****££(CL,UNIT
))£(CL,QQQQ)))
CL"COUNTER
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
CL"UNIT
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
EQ"IIIII"I"£(PS,END PROGRAM)£(PS,QQQQ*******££(CL,QQQQ))£(PS,COUNTER****££(CL,C
OUNTER))£(PS,UNIT*******££(CL,UNIT))"£(DS,UNIT,I££(CL,UNIT))£(PS,UNIT*****££(CL
,UNIT))£(CL,QQQQ)
CL"UNIT
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
DS"UNIT"II
CL"UNIT
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUUMENTS OF CL FUNCTION IGNORED+++++
PS"UNIT*****II
UNIT*****II
CL"QQQQ
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
PS"WHEN MUM1 MAM2 YOUM3 IN EARLY GREECE SHE SUNG
WHEN MUM1 MAM2 YOUM3 IN EARLY GREECE SHE SUNG
CL"QQQQ
```

```
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                2281
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2282
PS"QQQQ*****£(PS,WHEN MUM1 MAM2 YOUM3 IN EARLY GREECE SHE SUNG)£(PS,QQQQ******£$ 2283
(CL,QQQQ))£(SS,QQQQ,£(RS))£(DS,QQQQ,££(CL,QQQQ,£(RS)))£(EQ,££(CL,COUNTER),££(CL 2284
,UNIT),(£(PS,END PROGRAM)£(PS,QQQQ*******££(CL,QQQQ))£(PS,COUNTER****££(CL,COUN 2285
TER))£(PS,UNIT*******££(CL,UNIT))),(£(DS,UNIT,I££(CL,UNIT))£(PS,UNIT******££(CL, 2286
UNIT))£(CL,QQQQ)))                                                              2287
QQQQ*****£(PS,WHEN MUM1 MAM2 YOUM3 IN EARLY GREECE SHE SUNG)£(PS,QQQQ******££(CL 2288
,QQQQ))£(SS,QQQQ,£(RS))£(DS,QQQQ,££(CL,QQQQ,£(RS)))£(EQ,££(CL,COUNTER),££(CL,UN 2289
IT),(£(PS,END PROGRAM)£(PS,QQQQ*******££(CL,QQQQ))£(PS,COUNTER****££(CL,COUNTER 2290
))£(PS,UNIT*******££(CL,UNIT))),(£(DS,UNIT,I££(CL,UNIT))£(PS,UNIT******££(CL,UNI 2291
T))£(CL,QQQQ)))                                                                 2292
RS                                                                              2293
SS"QQQQ"M1"M2"M3                                                                2294
RS                                                                              2295
CL"QQQQ"MM1"MM2"MM3                                                             2296
DS"QQQQ"£(PS,WHEN MUMM1 MAMM2 YOUMM3 IN EARLY GREECE SHE SUNG)£(PS,QQQQ******££( 2297
CL,QQQQ))£(SS,QQQQ,£(RS))£(DS,QQQQ,££(CL,QQQQ,£(RS)))£(EQ,££(CL,COUNTER),££(CL, 2298
UNIT),(£(PS,END PROGRAM)£(PS,QQQQ*******££(CL,QQQQ))£(PS,COUNTER*****££(CL,COUNT 2299
ER))£(PS,UNIT*******££(CL,UNIT))),(£(DS,UNIT,I££(CL,UNIT))£(PS,UNIT******££(CL,U 2300
NIT))£(CL,QQQQ)))                                                               2301
CL"COUNTER                                                                      2302
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                2303
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2304
CL"UNIT                                                                         2305
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                2306
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2307
EQ"IIIII"II"£(PS,END PROGRAM)£(PS,QQQQ*******££(CL,QQQQ))£(PS,COUNTER*****££(CL, 2308
COUNTER))£(PS,UNIT*******££(CL,UNIT))"£(DS,UNIT,I££(CL,UNIT))£(PS,UNIT******££(C 2309
L,UNIT))£(CL,QQQQ)                                                              2310
CL"UNIT                                                                         2311
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                2312
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2313
DS"UNIT"III                                                                     2314
CL"UNIT                                                                         2315
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                2316
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2317
PS"UNIT*****III                                                                 2318
```

```
UNIT*****III                                                                    2319
CL"QQQQ                                                                         2320
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               2321
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2322
PS"WHEN MUMM1 MAMM2 YOUMM3 IN EARLY GREECE SHE SUNG                             2323
WHEN MUMM1 MAMM2 YOUMM3 IN EARLY GREECE SHE SUNG                                2324
CL"QQQQ                                                                         2325
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               2326
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2327
PS"QQQQ*****£(PS,WHEN MUMM1 MAMM2 YOUMM3 IN EARLY GREECE SHE SUNG)£(PS,QQQQ**** 2328
*££(CL,QQQQ))£(SS,QQQQ,£(RS))£(DS,QQQQ,££(CL,QQQQ,£(RS)))£(EQ,££(CL,COUNTER),†† 2329
(CL,UNIT),(£(PS,END PROGRAM)£(PS,QQQQ*******††(CL,QQQQ))£(PS,COUNTER*****††(CL,C 2330
OUNTER))£(PS,UNIT*******††(CL,UNIT))),(£(DS,UNIT,I††(CL,UNIT))£(PS,UNIT******††( 2331
CL,UNIT))£(CL,QQQQ)))                                                           2332
QQQQ*****£(PS,WHEN MUMM1 MAMM2 YOUMM3 IN EARLY GREECE SHE SUNG)£(PS,QQQQ******†† 2333
(CL,QQQQ))£(SS,QQQQ,£(RS))£(DS,QQQQ,££(CL,QQQQ,£(RS)))£(EQ,££(CL,COUNTER),††(CL 2334
,UNIT),(£(PS,END PROGRAM)£(PS,QQQQ*******††(CL,QQQQ))£(PS,COUNTER*****††(CL,COUN 2335
TER))£(PS,UNIT*******††(CL,UNIT))),(£(DS,UNIT,I††(CL,UNIT))£(PS,UNIT******††(CL, 2336
UNIT))£(CL,QQQQ)))                                                              2337
RS                                                                              2338
SS"QQQQ"MM1"MM2"MM3                                                             2339
RS                                                                              2340
CL"QQQQ"MMM1"MMM2"MMM3                                                          2341
DS"QQQQ"£(PS,WHEN MUMMM1 MAMMM2 YOUMMM3 IN EARLY GREECE SHE SUNG)£(PS,QQQQ***** 2342
††(CL,QQQQ))£(SS,QQQQ,£(RS))£(DS,QQQQ,††(CL,QQQQ,£(RS)))£(EQ,††(CL,COUNTER),††( 2343
CL,UNIT),(£(PS,END PROGRAM)£(PS,QQQQ*******††(CL,QQQQ))£(PS,COUNTER******††(CL,CO 2344
UNTER))£(PS,UNIT*******††(CL,UNIT))),(£(DS,UNIT,I††(CL,UNIT))£(PS,UNIT******††(C 2345
L,UNIT))£(CL,QQQQ)))                                                            2346
CL"COUNTER                                                                      2347
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               2348
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2349
CL"UNIT                                                                         2350
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               2351
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2352
EQ"IIIII"III"£(PS,END PROGRAM)£(PS,QQQQ*******††(CL,QQQQ))£(PS,COUNTER*****††(CL 2353
,COUNTER))£(PS,UNIT*******††(CL,UNIT))"£(DS,UNIT,I††(CL,UNIT))£(PS,UNIT******††( 2354
CL,UNIT))£(CL,QQQQ)                                                             2355
CL"UNIT                                                                         2356
```

```
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                2357
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2358
DS"UNIT"IIII                                                                    2359
CL"UNIT                                                                         2360
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                2361
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2362
PS"UNIT*****IIII                                                                2363
UNIT*****IIII                                                                   2364
CL"QQQQ                                                                         2365
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                2366
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2367
PS"WHEN MUMMM1 MAMMM2 YOUMMM3 IN EARLY GREECE SHE SUNG                          2368
WHEN MUMMM1 MAMMM2 YOUMMM3 IN EARLY GREECE SHE SUNG                             2369
CL"QQQQ                                                                         2370
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                2371
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2372
PS"QQQQ*****£(PS,WHEN MUMMM1 MAMMM2 YOUMMM3 IN EARLY GREECE SHE SUNG)£(PS,QQQQ* 2373
****££(CL,QQQQ))£(SS,QQQQ,£(RS))£(DS,QQQQ,££(CL,QQQQ,£(RS)))£(EQ,££(CL,COUNTER) 2374
,££(CL,UNIT),(£(PS,END PROGRAM)£(PS,QQQQ*******££(CL,QQQQ))£(PS,COUNTER*****£(C 2375
L,COUNTER))£(PS,UNIT*******££(CL,UNIT))),(£(DS,UNIT,I££(CL,UNIT))£(PS,UNIT***** 2376
££(CL,UNIT))£(CL,QQQQ)))                                                        2377
QQQQ*****£(PS,WHEN MUMMM1 MAMMM2 YOUMMM3 IN EARLY GREECE SHE SUNG)£(PS,QQQQ**** 2378
*££(CL,QQQQ))£(SS,QQQQ,£(RS))£(DS,QQQQ,££(CL,QQQQ,£(RS)))£(EQ,££(CL,COUNTER),†† 2379
(CL,UNIT),(£(PS,END PROGRAM)£(PS,QQQQ*******££(CL,QQQQ))£(PS,COUNTER*****£(CL,C 2380
OUNTER))£(PS,UNIT*******££(CL,UNIT))),(£(DS,UNIT,I££(CL,UNIT))£(PS,UNIT******£( 2381
CL,UNIT))£(CL,QQQQ)))                                                           2382
RS                                                                              2383
SS"QQQQ"MMM1"MMM2"MMM3                                                          2384
RS                                                                              2385
CL"QQQQ"MMMM1"MMMM2"MMMM3                                                       2386
DS"QQQQ"£(PS,WHEN MUMMMM1 MAMMMM2 YOUMMMM3 IN EARLY GREECE SHE SUNG)£(PS,QQQQ** 2387
***££(CL,QQQQ))£(SS,QQQQ,£(RS))£(DS,QQQQ,££(CL,QQQQ,£(RS)))£(EQ,££(CL,COUNTER), 2388
££(CL,UNIT),(£(PS,END PROGRAM)£(PS,QQQQ*******££(CL,QQQQ))£(PS,COUNTER******£(CL 2389
,COUNTER))£(PS,UNIT*******££(CL,UNIT))),(£(DS,UNIT,I££(CL,UNIT))£(PS,UNIT*****£ 2390
£(CL,UNIT))£(CL,QQQQ)))                                                         2391
CL"COUNTER                                                                      2392
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                2393
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2394
```

```
CL"UNIT                                                                    2395
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                          2396
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++           2397
EQ"IIIII"IIII"£(PS,END PROGRAM)£(PS,QQQQ*******££(CL,QQQQ))£(PS,COUNTER****££(C 2398
L,COUNTER))£(PS,UNIT*******££(CL,UNIT))"£(DS,UNIT,I££(CL,UNIT))£(PS,UNIT******£ 2399
(CL,UNIT))£(CL,QQQQ)                                                       2400
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                          2401
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++           2402
DS"UNIT"IIIII                                                              2403
CL"UNIT                                                                    2404
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                          2405
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++           2406
PS"UNIT*****IIIII                                                          2407
UNIT*****IIIII                                                             2408
CL"QQQQ                                                                    2409
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                          2410
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++           2411
PS"WHEN MUMMM1 MAMMM2 YOUMMM3 IN EARLY GREECE SHE SUNG                     2412
WHEN MUMMM1 MAMMM2 YOUMMM3 IN EARLY GREECE SHE SUNG                        2413
CL"QQQQ                                                                    2414
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                          2415
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++           2416
PS"QQQQ*****£(PS,WHEN MUMMM1 MAMMM2 YOUMMM3 IN EARLY GREECE SHE SUNG)£(PS,QQ 2417
QQ*****££(CL,QQQQ))£(SS,QQQQ,£(RS))£(DS,QQQQ,££(CL,QQQQ,£(RS)))£(EQ,££(CL,COUNT 2418
ER),££(CL,UNIT),(£(PS,END PROGRAM)£(PS,QQQQ*******££(CL,QQQQ))£(PS,COUNTER****£ 2419
£(CL,COUNTER))£(PS,UNIT*******££(CL,UNIT))),(£(DS,UNIT,I££(CL,UNIT))£(PS,UNIT** 2420
***££(CL,UNIT))£(CL,QQQQ)))                                                2421
QQQQ*****£(PS,WHEN MUMMM1 MAMMM2 YOUMMM3 IN EARLY GREECE SHE SUNG)£(PS,QQQQ* 2422
****££(CL,QQQQ))£(SS,QQQQ,£(RS))£(DS,QQQQ,££(CL,QQQQ,£(RS)))£(EQ,££(CL,COUNTER) 2423
,££(CL,UNIT),(£(PS,END PROGRAM)£(PS,QQQQ*******££(CL,QQQQ))£(PS,COUNTER****££(C 2424
L,COUNTER))£(PS,UNIT*******££(CL,UNIT))),(£(DS,UNIT,I££(CL,UNIT))£(PS,UNIT***** 2425
££(CL,UNIT))£(CL,QQQQ)))                                                   2426
RS                                                                         2427
SS"QQQQ"MMM1"MMM2"MMM3                                                     2428
RS                                                                         2429
CL"QQQQ"SIC,HEAVENLY"ID,WAS"NG,WHILE YET                                   2430
DS"QQQQ"£(PS,WHEN MUSIC,HEAVENLY MAID,WAS YOUNG,WHILE YET IN EARLY GREECE SHE S 2431
UNG)£(PS,QQQQ*****££(CL,QQQQ))£(SS,QQQQ,£(RS))£(DS,QQQQ,££(CL,QQQQ,£(RS)))£(EQ, 2432
```

```
&&(CL,COUNTER),&&(CL,UNIT),(&(PS,END PROGRAM)&(PS,QQQQ*******&&(CL,QQQQ))&(PS,C 2433
OUNTER****&&(CL,COUNTER))&(PS,UNIT*******&&(CL,UNIT))),(&(DS,UNIT,I&&(CL,UNIT)) 2434
&(PS,UNIT******&&(CL,UNIT))&(CL,QQQQ))) 2435
CL"COUNTER 2436
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++ 2437
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++ 2438
CL"UNIT 2439
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++ 2440
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++ 2441
EQ"IIIII"IIIII"&(PS,END PROGRAM)&(PS,QQQQ*******&&(CL,QQQQ))&(PS,COUNTER****&&( 2442
CL,COUNTER))&(PS,UNIT*******&&(CL,UNIT))"&(DS,UNIT,I&&(CL,UNIT))&(PS,UNIT******& 2443
&(CL,UNIT))&(CL,QQQQ) 2444
PS"END PROGRAM 2445
END PROGRAM 2446
CL"QQQQ 2447
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++ 2448
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++ 2449
PS"QQQQ*******&(PS,WHEN MUSIC,HEAVENLY MAID,WAS YOUNG,WHILE YET IN EARLY GREECE 2450
 SHE SUNG)&(PS,QQQQ*****&&(CL,QQQQ))&(SS,QQQQ,&(RS))&(DS,QQQQ,&&(CL,QQQQ,&(RS)) 2451
)&(EQ,&&(CL,COUNTER),&&(CL,UNIT),(&(PS,END PROGRAM)&(PS,QQQQ*******&&(CL,QQQQ)) 2452
&(PS,COUNTER****&&(CL,COUNTER))&(PS,UNIT*******&&(CL,UNIT))),(&(DS,UNIT,I&&(CL, 2453
UNIT))&(PS,UNIT******&&(CL,UNIT))&(CL,QQQQ))) 2454
QQQQ*******&(PS,WHEN MUSIC,HEAVENLY MAID,WAS YOUNG,WHILE YET IN EARLY GREECE SH 2455
E SUNG)&(PS,QQQQ*****&&(CL,QQQQ))&(SS,QQQQ,&(RS))&(DS,QQQQ,&&(CL,QQQQ,&(RS)))&( 2456
EQ,&&(CL,COUNTER),&&(CL,UNIT),(&(PS,END PROGRAM)&(PS,QQQQ*******&&(CL,QQQQ))&(P 2457
S,COUNTER****&&(CL,COUNTER))&(PS,UNIT*******&&(CL,UNIT))),(&(DS,UNIT,I&&(CL,UNI 2458
T))&(PS,UNIT******&&(CL,UNIT))&(CL,QQQQ))) 2459
CL"COUNTER 2460
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++ 2461
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++ 2462
PS"COUNTER****IIIII 2463
COUNTER****IIIII 2464
CL"UNIT 2465
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++ 2466
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++ 2467
PS"UNIT******IIIII 2468
UNIT******IIIII 2469
PS" 2470
```

```
+++++THIS PS FUNCTION HAS NULL STRING+++++
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++
RS
CL"QQQQ
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
PS"WHEN MUSIC"HEAVENLY MAID"WAS YOUNG"WHILE YET IN EARLY GREECE SHE SUNG
WHEN MUSIC"HEAVENLY MAID"WAS YOUNG"WHILE YET IN EARLY GREECE SHE SUNG
CL"QQQQ
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
PS"QQQQ*****£(PS,WHEN MUSIC,HEAVENLY MAID,WAS YOUNG,WHILE YET IN EARLY GREECE S
HE SUNG)£(PS,QQQQ*****££(CL,QQQQ))£(SS,QQQQ,£(RS))£(DS,QQQQ,££(CL,QQQQ,£(RS)))£
(EQ,££(CL,COUNTER),££(CL,UNIT),(£(PS,END PROGRAM)£(PS,QQQQ*******££(CL,QQQQ))£(
(EQ,££(CL,COUNTER),££(CL,UNIT),(£(PS,END PROGRAM)£(PS,QQQQ*******££(CL,QQQQ))£(
PS,COUNTER*****££(CL,COUNTER))£(PS,UNIT*******££(CL,UNIT))),(£(DS,UNIT,I££(CL,UN
IT))£(PS,UNIT******££(CL,UNIT))£(CL,QQQQ)))
QQQQ*****£(PS,WHEN MUSIC,HEAVENLY MAID,WAS YOUNG,WHILE YET IN EARLY GREECE SHE
SUNG)£(PS,QQQQ*****££(CL,QQQQ))£(SS,QQQQ,£(RS))£(DS,QQQQ,££(CL,QQQQ,£(RS)))£(EQ
,££(CL,COUNTER),££(CL,UNIT),(£(PS,END PROGRAM)£(PS,QQQQ*******££(CL,QQQQ))£(PS,
COUNTER*****££(CL,COUNTER))£(PS,UNIT*******££(CL,UNIT))),(£(DS,UNIT,I££(CL,UNIT)
)£(PS,UNIT******££(CL,UNIT))£(CL,QQQQ)))
RS
PS"---------------------------------------------------------- - - -
------------------------------------------------------------------
SS"QQQQ"
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++SS"QQQQ"
RS


+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++
------------------------------------------------------------------
**********TEST-5 EQ FUNCTION**********
+++++THIS PS FUNCTION HAS NULL STRING+++++
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
```

```
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    2509
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     2510
THIS IS A CAT                                                       2511
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    2512
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     2513
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    2514
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     2515
THIS IS A DOG                                                       2516
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    2517
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     2518
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    2519
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     2520
THIS IS A MOUSE                                                     2521
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    2522
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     2523
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    2524
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     2525
THIS IS A SHEEP                                                     2526
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    2527
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     2528
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    2529
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     2530
THIS IS A COW                                                       2531
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    2532
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     2533
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    2534
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     2535
XYZ*******THIS IS A HORSE                                           2536
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    2537
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     2538
END-PROC**THIS IS A HORSE                                           2539
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    2540
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     2541
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    2542
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     2543
THIS IS A CAT                                                       2544
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    2545
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     2546
```

```
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    2547
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     2548
THIS IS A DOG                                                       2549
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    2550
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     2551
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    2552
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     2553
THIS IS A MOUSE                                                     2554
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    2555
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     2556
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    2557
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     2558
THIS IS A SHEEP                                                     2559
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    2560
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     2561
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    2562
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     2563
THIS IS A COW                                                       2564
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    2565
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     2566
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    2567
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     2568
END OF PROGRAM                                                      2569
NAME**********XYZ                                                  2570
NAME**********END-PROC                                           2571
NAME**********ZYX                                                  2572
NAME**********OTHER                                                2573
NAME**********ALTER                                                2574
+++++THERE ARE NO FORMS IN STORE+++++                               2575
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                   2576
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++    2577
                                                                    2578
                                                                    2579
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                       2580
-------------------------------------------------------------------- 2581
**********TEST-5 EQ FUNCTION**WITH TRACE************************** 2582
PS"                                                                 2583
+++++THIS PS FUNCTION HAS NULL STRING+++++                          2584
```

```
+++++TC END TRAC JCB ,TYPE ' IN CCLUMN 1+++++
RS
RS
DS"XYZ"THIS IS A CAT
RS
SS"XYZ"CAT
RS
DS"END-PROC"THIS IS A HORSE
RS
RS
DS"ZYX"THIS IS A CAT
RS
RS
SS"ZYX"CAT
DS"OTHER"£(EQ,££(CL,ZYX,£(RS)),££(CL,END-PRCC),(£(PS,END CF PRCGRAM)£(LN,NAME**
********)),(£(PS,££(CL,ZYX,£(RS)))))£(CL,CTHER)
DS"ALTER"£(EQ,££(CL,XYZ,£(RS)),££(CL,END-PRCC),(£(PS,XYZ*******££(CL,XYZ,£(RS))
)£(PS,END-PRCC**££(CL,END-PRCC))£(CL,CTHER)),(£(PS,££(CL,XYZ,£(RS)))£(CL,ALTER)
))
CL"ALTER
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++
+++++FCRM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
RS
CL"XYZ"CAT
CL"END-PRCC
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++
+++++FCRM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTICN IGNORED+++++
EQ"THIS IS A CAT"THIS IS A HORSE"£(PS,XYZ*******££(CL,XYZ,£(RS)))£(PS,END-PRCC*
*££(CL,END-PRCC))£(CL,CTHER)"£(PS,££(CL,XYZ,£(RS)))£(CL,ALTER)
RS
CL"XYZ"CAT
PS"THIS IS A CAT
THIS IS A CAT
CL"ALTER
+++++THERE ARE NC ARGUMENTS IN THAT FUNCTICN+++++
+++++FCRM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTICN IGNORED+++++
RS
CL"XYZ"DCG
```

```
CL"END-PROC                                                                    2623
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              2624
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++               2625
EQ"THIS IS A DOG"THIS IS A HORSE"£(PS,XYZ*******££(CL,XYZ,£(RS)))£(PS,END-PROC* 2626
*££(CL,END-PROC))£(CL,OTHER)"£(PS,££(CL,XYZ,£(RS)))£(CL,ALTER)                2627
RS                                                                             2628
CL"XYZ"DOG                                                                     2629
PS"THIS IS A DOG                                                               2630
THIS IS A DOG                                                                  2631
CL"ALTER                                                                       2632
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              2633
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++               2634
RS                                                                             2635
CL"XYZ"MOUSE                                                                   2636
CL"END-PROC                                                                    2637
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              2638
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++               2639
EQ"THIS IS A MOUSE"THIS IS A HORSE"£(PS,XYZ*******££(CL,XYZ,£(RS)))£(PS,END-PRO 2640
C**££(CL,END-PROC))£(CL,OTHER)"£(PS,££(CL,XYZ,£(RS)))£(CL,ALTER)              2641
RS                                                                             2642
                                                                               2643
CL"XYZ"MOUSE                                                                   2644
PS"THIS IS A MOUSE                                                             2645
THIS IS A MOUSE                                                                2646
CL"ALTER                                                                       2647
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              2648
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++               2649
RS                                                                             2650
CL"XYZ"SHEEP                                                                   2651
CL"END-PROC                                                                    2652
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              2653
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++               2654
EQ"THIS IS A SHEEP"THIS IS A HORSE"£(PS,XYZ*******££(CL,XYZ,£(RS)))£(PS,END-PRO 2655
C**££(CL,END-PROC))£(CL,OTHER)"£(PS,££(CL,XYZ,£(RS)))£(CL,ALTER)              2656
RS                                                                             2657
CL"XYZ"SHEEP                                                                   2658
PS"THIS IS A SHEEP                                                             2659
THIS IS A SHEEP                                                                2660
```

```
CL"ALTER                                                                    2661
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                           2662
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++            2663
RS                                                                          2664
CL"XYZ"COW                                                                  2665
CL"END-PROC                                                                 2666
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                           2667
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++            2668
EQ"THIS IS A COW"THIS IS A HORSE"£(PS,XYZ*******££(CL,XYZ,£(RS)))£(PS,END-PROC* 2669
*££(CL,END-PROC))£(CL,OTHER)"£(PS,££(CL,XYZ,£(RS)))£(CL,ALTER)             2670
RS                                                                          2671
CL"XYZ"COW                                                                  2672
PS"THIS IS A COW                                                            2673
THIS IS A COW                                                               2674
CL"ALTER                                                                    2675
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                           2676
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++            2677
RS                                                                          2678
CL"XYZ"HORSE                                                                2679
CL"END-PROC                                                                 2680
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                           2681
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++            2682
EQ"THIS IS A HORSE"THIS IS A HORSE"£(PS,XYZ*******££(CL,XYZ,£(RS)))£(PS,END-PRO 2683
C**££(CL,END-PROC))£(CL,OTHER)"£(PS,££(CL,XYZ,£(RS)))£(CL,ALTER)           2684
RS                                                                          2685
CL"XYZ"HORSE                                                                2686
PS"XYZ*******THIS IS A HORSE                                                2687
XYZ*******THIS IS A HORSE                                                   2688
CL"END-PROC                                                                 2689
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                           2690
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++            2691
PS"END-PROC**THIS IS A HORSE                                                2692
END-PROC**THIS IS A HORSE                                                   2693
CL"OTHER                                                                    2694
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                           2695
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++            2696
RS                                                                          2697
CL"ZYX"CAT                                                                  2698
```

```
CL"END-PROC                                                                    2699
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              2700
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++               2701
EQ"THIS IS A CAT"THIS IS A HORSE"£(PS,END OF PROGRAM)£(LN,NAME**********)"£(PS, 2702
££(CL,ZYX,£(RS)))                                                              2703
RS                                                                             2704
CL"ZYX"CAT                                                                     2705
PS"THIS IS A CAT                                                               2706
THIS IS A CAT                                                                  2707
CL"OTHER                                                                       2708
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              2709
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++               2710
RS                                                                             2711
CL"ZYX"DOG                                                                     2712
CL"END-PROC                                                                    2713
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              2714
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++               2715
EQ"THIS IS A DOG"THIS IS A HORSE"£(PS,END OF PROGRAM)£(LN,NAME**********)"£(PS, 2716
££(CL,ZYX,£(RS)))                                                              2717
RS                                                                             2718
CL"ZYX"DOG                                                                     2719
PS"THIS IS A DOG                                                               2720
THIS IS A DOG                                                                  2721
CL"OTHER                                                                       2722
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              2723
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++               2724
RS                                                                             2725
CL"ZYX"MOUSE                                                                   2726
CL"END-PROC                                                                    2727
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                              2728
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++               2729
EQ"THIS IS A MOUSE"THIS IS A HORSE"£(PS,END OF PROGRAM)£(LN,NAME**********)"£(P 2730
S,££(CL,ZYX,£(RS)))                                                            2731
RS                                                                             2732
CL"ZYX"MOUSE                                                                   2733
PS"THIS IS A MOUSE                                                             2734
THIS IS A MOUSE                                                                2735
CL"OTHER                                                                       2736
```

```
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                2737
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2738
RS                                                                              2739
CL"ZYX"SHEEP                                                                    2740
CL"END-PROC                                                                     2741
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                2742
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2743
EQ"THIS IS A SHEEP"THIS IS A HORSE"£(PS,END OF PROGRAM)£(LN,NAME**********)"£(P 2744
S,$$(CL,ZYX,£(RS)))                                                             2745
RS                                                                              2746
CL"ZYX"SHEEP                                                                    2747
PS"THIS IS A SHEEP                                                              2748
THIS IS A SHEEP                                                                 2749
CL"OTHER                                                                        2750
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                2751
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2752
RS                                                                              2753
CL"ZYX"CCW                                                                      2754
CL"END-PROC                                                                     2755
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                2756
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2757
EQ"THIS IS A CCW"THIS IS A HORSE"£(PS,END OF PROGRAM)£(LN,NAME**********)"£(PS, 2758
$$(CL,ZYX,£(RS)))                                                               2759
RS                                                                              2760
CL"ZYX"COW                                                                      2761
PS"THIS IS A CCW                                                                2762
THIS IS A CCW                                                                   2763
CL"OTHER                                                                        2764
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                2765
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2766
RS                                                                              2767
CL"ZYX"HORSE                                                                    2768
CL"END-PROC                                                                     2769
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                2770
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2771
EQ"THIS IS A HORSE"THIS IS A HORSE"£(PS,END OF PROGRAM)£(LN,NAME**********)"£(P 2772
S,$$(CL,ZYX,£(RS)))                                                             2773
PS"END OF PROGRAM                                                               2774
```

```
END OF PROGRAM                                                          2775
LN"NAME**********                                                       2776
NAME**********XYZ                                                      2777
NAME**********END-PROC                                                 2778
NAME**********ZYX                                                      2779
NAME**********OTHER                                                    2780
NAME**********ALTER                                                    2781
+++++THERE ARE NO FORMS IN STORE+++++                                   2782
CL"OTHER                                                                2783
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                       2784
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++        2785
RS                                                                      2786
                                                                        2787
                                                                        2788
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                           2789
------------------------------------------------------------------------2790
**********TEST-6 EQ FUNCTION**************************************** 2791
+++++THIS PS FUNCTION HAS NULL STRING+++++                              2792
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                           2793
+++++THIS PS FUNCTION HAS NULL STRING+++++                              2794
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                           2795
+++++THIS PS FUNCTION HAS NULL STRING+++++                              2796
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                           2797
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                       2798
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++        2799
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                       2800
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++        2801
TEST*******=A**2 -2*A*B +B**2                                           2802
EXPRESSION*=A**2 +2*A*B +B**2                                           2803
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                           2804
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                       2805
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++        2806
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                       2807
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++        2808
TEST*******=A**2 -2*A*B +B**2                                           2809
EXPRESSION*=A**2 +2*A*B -B**2                                           2810
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                           2811
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                       2812
```

```
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
TEST********=A**2 -2*A*B +B**2
EXPRESSION*=A**2 -2*A*B -B**2
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
END PROGRAM
EXPRESSION*=A**2 -2*A*B +B**2
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
TEST********=A**2 -2*A*B +B**2
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++


+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++
------------------------------------------------------------------------
**********TEST-6 EQ FUNCTION**WITH TRACE**************************
PS"
+++++THIS PS FUNCTION HAS NULL STRING+++++
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++
RS
DS"EXPRESSION"AEXPONENT2 SIGN12TIMESATIMESB SIGN2BEXPONENT2
PS"
+++++THIS PS FUNCTION HAS NULL STRING+++++
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++
RS
DS"TEST"A**2 -2*A*B +B**2
PS"
+++++THIS PS FUNCTION HAS NULL STRING+++++
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++
RS
SS"EXPRESSION"EXPONENT"TIMES"SIGN1"SIGN2
PS"
+++++THIS PS FUNCTION HAS NULL STRING+++++
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++
RS
```

```
CL"TEST                                                                         2851
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               2852
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2853
RS                                                                              2854
RS                                                                              2855
CL"EXPRESSION"**"*"+"+                                                          2856
EQ"A**2 -2*A*B +B**2"A**2 +2*A*B +B**2"£(PS,END PROGRAM)£(PS,EXPRESSION*=££(CL, 2857
EXPRESSION,**,*,£(RS),£(RS)))£(PS,TEST*******=££(CL,TEST))"£(PS,EXPRESSION*=££( 2858
CL,EXPRESSION,**,*,£(RS),£(RS))£(PS,TEST*******=££(CL,TEST)))                   2859
RS                                                                              2860
RS                                                                              2861
CL"EXPRESSION"**"*"+"+                                                          2862
CL"TEST                                                                         2863
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               2864
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2865
PS"TEST*******=A**2 -2*A*B +B**2                                                2866
TEST*******=A**2 -2*A*B +B**2                                                   2867
PS"EXPRESSION*=A**2 +2*A*B +B**2                                                2868
EXPRESSION*=A**2 +2*A*B +B**2                                                   2869
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                   2870
RS                                                                              2871
CL"TEST                                                                         2872
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               2873
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2874
RS                                                                              2875
RS                                                                              2876
CL"EXPRESSION"**"*"+"                                                           2877
EQ"A**2 -2*A*B +B**2"A**2 +2*A*B  B**2"£(PS,END PROGRAM)£(PS,EXPRESSION*=££(CL, 2878
EXPRESSION,**,*,£(RS),£(RS)))£(PS,TEST*******=££(CL,TEST))"£(PS,EXPRESSION*=££( 2879
CL,EXPRESSION,**,*,£(RS),£(RS))£(PS,TEST*******=££(CL,TEST)))                   2880
RS                                                                              2881
RS                                                                              2882
CL"EXPRESSION"**"*"+"-                                                          2883
CL"TEST                                                                         2884
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               2885
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2886
PS"TEST*******=A**2 -2*A*B +B**2                                                2887
TEST*******=A**2 -2*A*B +B**2                                                   2888
```

```
PS"EXPRESSION*=A**2 +2*A*B -B**2                                                2889
EXPRESSION*=A**2 +2*A*B -B**2                                                   2890
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++                                   2891
RS                                                                              2892
CL"TEST                                                                         2893
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               2894
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2895
RS                                                                              2896
RS                                                                              2897
CL"EXPRESSION"**"*"-"-                                                          2898
EQ"A**2 -2*A*B +B**2"A**2 -2*A*B -B**2"£(PS,END PROGRAM)£(PS,EXPRESSION*=££(CL, 2899
EXPRESSION,**,*,£(RS),£(RS)))£(PS,TEST*******=££(CL,TEST))"£(PS,EXPRESSION*=££( 2900
CL,EXPRESSION,**,*,£(RS),£(RS))£(PS,TEST*******=££(CL,TEST)))                   2901
RS                                                                              2902
RS                                                                              2903
CL"EXPRESSION"**"*"-"-                                                          2904
CL"TEST                                                                         2905
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               2906
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2907
PS"TEST*******=A**2 -2*A*B +B**2                                                2908
TEST*******=A**2 -2*A*B +B**2                                                   2909
PS"EXPRESSION*=A**2 -2*A*B -B**2                                                2910
EXPRESSION*=A**2 -2*A*B -B**2                                                   2911
+++++TC END TRAC JCB ,TYPE ' IN COLUMN 1+++++                                   2912
RS                                                                              2913
CL"TEST                                                                         2914
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                               2915
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                 2916
RS                                                                              2917
RS                                                                              2918
CL"EXPRESSION"**"*"-"+                                                          2919
EQ"A**2 -2*A*B +B**2"A**2 -2*A*B +B**2"£(PS,END PROGRAM)£(PS,EXPRESSION*=££(CL, 2920
EXPRESSION,**,*,£(RS),£(RS)))£(PS,TEST*******=££(CL,TEST))"£(PS,EXPRESSION*=££( 2921
CL,EXPRESSION,**,*,£(RS),£(RS))£(PS,TEST*******=££(CL,TEST)))                   2922
PS"END PROGRAM                                                                  2923
END PROGRAM                                                                     2924
RS                                                                              2925
RS                                                                              2926
```

```
CL"EXPRESSION"**"*"-"+                                                    2927
PS"EXPRESSION*=A**2 -2*A*B +B**2                                          2928
EXPRESSION*=A**2 -2*A*B +B**2                                             2929
CL"TEST                                                                   2930
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                         2931
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++          2932
PS"TEST*******=A**2 -2*A*B +B**2                                          2933
TEST*******=A**2 -2*A*B +B**2                                             2934
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                             2935
RS                                                                        2936
                                                                          2937
                                                                          2938
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                             2939
------------------------------------------------------------------------  2940
**********TEST-7 EQ FUNCTION**************************************        2941
+++++THIS PS FUNCTION HAS NULL STRING+++++                                2942
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                             2943
+++++THIS PS FUNCTION HAS NULL STRING+++++                                2944
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                             2945
+++++THIS PS FUNCTION HAS NULL STRING+++++                                2946
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                             2947
+++++THIS PS FUNCTION HAS NULL STRING+++++                                2948
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                             2949
+++++THIS PS FUNCTION HAS NULL STRING+++++                                2950
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                             2951
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                         2952
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++          2953
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                         2954
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++          2955
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                         2956
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++          2957
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                         2958
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++          2959
A WISE MAN WILL MAKE MORE OPPORTUNITIES THAN HE FINDS                     2960
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                         2961
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++          2962
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                         2963
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++          2964
```

```
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    2965
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     2966
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    2967
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     2968
MONEY IS LIKE MUCK , NOT GOOD EXCEPT IT CAN BE SPREAD               2969
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    2970
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     2971
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    2972
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     2973
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    2974
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     2975
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    2976
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     2977
THE REMEDY IS WORSE THAN THE DISEASE                                2978
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    2979
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     2980
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    2981
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     2982
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    2983
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     2984
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    2985
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     2986
STUDIES SERVE FOR DELIGHT,FOR ORNAMENT,AND FOR ABILITY              2987
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    2988
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     2989
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    2990
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     2991
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    2992
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     2993
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                    2994
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++     2995
NAME  P1                                                            2996
NAME  END                                                           2997
NAME  X                                                             2998
NAME  Y                                                             2999
NAME  A                                                             3000
+++++THERE ARE NO FORMS IN STORE+++++                               3001
END OF PROGRAM                                                      3002
```

```
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                        3003
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++         3004
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                        3005
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++         3006
                                                                        3007
                                                                        3008
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                           3009
------------------------------------------------------------------------3010
**********TEST-7 EQ FUNCTION**WITH TRACE**************************     3011
PS"                                                                     3012
**********TEST-7 EQ FUNCTION**WITH TRACE**************************     3013
PS"                                                                     3014
+++++THIS PS FUNCTION HAS NULL STRING+++++                              3015
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                           3016
RS                                                                      3017
DS"P1"£(PS,£(RS))                                                       3018
PS"                                                                     3019
+++++THIS PS FUNCTION HAS NULL STRING+++++                              3020
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                           3021
RS                                                                      3022
DS"END"£(LN,NAME  )£(PS,END OF PROGRAM)                                 3023
PS"                                                                     3024
+++++THIS PS FUNCTION HAS NULL STRING+++++                              3025
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                           3026
RS                                                                      3027
DS"X"EQ,£(RS),END-PR,                                                   3028
PS"                                                                     3029
+++++THIS PS FUNCTION HAS NULL STRING+++++                              3030
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                           3031
RS                                                                      3032
DS"Y"(£(CL,END)),(£(CL,P1))                                             3033
PS"                                                                     3034
+++++THIS PS FUNCTION HAS NULL STRING+++++                              3035
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                           3036
RS                                                                      3037
DS"A"£(£(CL,X)£(CL,Y))£(CL,A)                                           3038
PS"                                                                     3039
+++++THIS PS FUNCTION HAS NULL STRING+++++                              3040
```

```
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                    3041
RS                                                               3042
CL"A                                                             3043
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                3044
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++ 3045
CL"X                                                             3046
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                3047
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++ 3048
RS                                                               3049
CL"Y                                                             3050
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                3051
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++ 3052
EQ"FIRST EXECUTION"END-PR"£(CL,END)"£(CL,P1)                     3053
CL"P1                                                            3054
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                3055
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++ 3056
RS                                                               3057
PS"A WISE MAN WILL MAKE MORE OPPORTUNITIES THAN HE FINDS         3058
A WISE MAN WILL MAKE MORE OPPORTUNITIES THAN HE FINDS            3059
CL"A                                                             3060
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                3061
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++ 3062
CL"X                                                             3063
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                3064
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++ 3065
RS                                                               3066
CL"Y                                                             3067
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                3068
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++ 3069
EQ"SECOND EXECUTION"END-PR"£(CL,END)"£(CL,P1)                    3070
CL"P1                                                            3071
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                3072
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++ 3073
RS                                                               3074
PS"MONEY IS LIKE MUCK , NOT GOOD EXCEPT IT CAN BE SPREAD         3075
MONEY IS LIKE MUCK , NOT GOOD EXCEPT IT CAN BE SPREAD            3076
CL"A                                                             3077
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                3078
```

```
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
CL"X
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
RS
CL"Y
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
EQ"THIRD EXECUTION"END-PR"£(CL,END)"£(CL,P1)
CL"P1
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
RS
PS"THE REMEDY IS WORSE THAN THE DISEASE
THE REMEDY IS WORSE THAN THE DISEASE
CL"A
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
CL"X
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
RS
CL"Y
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
EQ"FOURTH EXECUTION"END-PR"£(CL,END)"£(CL,P1)
CL"P1
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
RS
PS"STUDIES SERVE FOR DELIGHT,FOR ORNAMENT,AND FOR ABILITY
STUDIES SERVE FOR DELIGHT,FOR ORNAMENT,AND FOR ABILITY
CL"A
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
CL"X
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
```

```
RS                                                                    3117
CL"Y                                                                  3118
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                     3119
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++      3120
EQ"END-PR"END-PR"£(CL,END)"£(CL,P1)                                   3121
CL"END                                                                3122
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                     3123
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++      3124
LN"NAME                                                               3125
NAME  P1                                                              3126
NAME  END                                                             3127
NAME  X                                                               3128
NAME  Y                                                               3129
NAME  A                                                               3130
+++++THERE ARE NO FORMS IN STORE+++++                                 3131
PS"END OF PROGRAM                                                     3132
END OF PROGRAM                                                        3133
CL"A                                                                  3134
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                     3135
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++      3136
CL"X                                                                  3137
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                     3138
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++      3139
RS                                                                    3140
                                                                      3141
                                                                      3142
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                         3143
----------------------------------------------------------------------3144
**********TEST-8 EQ FUNCTION**************************************** 3145
+++++THIS PS FUNCTION HAS NULL STRING+++++                            3146
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++                         3147
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                     3148
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++      3149
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                     3150
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++      3151
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                     3152
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++      3153
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                     3154
```

```
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
+++++NO SEGMENTATION OF THAT FORM HAS OCCURED+++++
+++++SS"S"ABC
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++
AC
+++++THIS PS FUNCTION HAS NULL STRING+++++
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++
-------------------------------------------------------------------
+++++THIS PS FUNCTION HAS NULL STRING+++++
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++


+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++
-------------------------------------------------------------------
**********TEST-8 EQ FUNCTION**WITH  TRACE**********
PS"
+++++THIS PS FUNCTION HAS NULL STRING+++++
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++
RS
DS"Q"£(DS,S,X)£(SS,S,Y)£(DS,Z,££(CL,S,A))£(EQ,££(CL,S,Y),££(CL,Z),££(CL,Z),(£(C
L,Q,££(CL,Z),Y)))
SS"Q"X"Y
PS"
+++++THIS PS FUNCTION HAS NULL STRING+++++
+++++TC END TRAC JOB ,TYPE ' IN COLUMN 1+++++
RS
LN"NAME
NAME  Q
+++++THERE ARE NO FORMS IN STORE+++++
```

```
PF"Q                                                                                                3193
£(DS,S,&1)£(SS,S,&2)£(DS,Z,££(CL,S,A))£(EQ,££(CL,S,&2),££(CL,Z),££(CL,Z),(£(CL,                     3194
Q,££(CL,Z),&2)))                                                                                    3195
PF"S                                                                                                3196
+++++ THERE IS NOT SUCH A FORM IN STORE+++++                                                        3197
+++++PF"S                                                                                           3198
PF"Z                                                                                                3199
+++++ THERE IS NOT SUCH A FORM IN STORE+++++                                                        3200
+++++PF"Z                                                                                           3201
PS"                                                                                                 3202
+++++THIS PS FUNCTION HAS NULL STRING+++++                                                          3203
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                                                       3204
RS                                                                                                  3205
CL"Q"ABCBCC"ABC                                                                                     3206
DS"S"ABCBCC                                                                                         3207
SS"S"ABC                                                                                            3208
CL"S"A                                                                                              3209
DS"Z"ABCC                                                                                           3210
CL"S"ABC                                                                                            3211
CL"Z                                                                                                3212
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                                   3213
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                                    3214
CL"Z                                                                                                3215
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                                   3216
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                                    3217
EQ"ABCBCC"ABCC"ABCC"£(CL,Q,££(CL,Z),ABC)                                                            3218
CL"Z                                                                                                3219
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                                   3220
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                                    3221
CL"Q"ABCC"ABC                                                                                       3222
DS"S"ABCC                                                                                           3223
SS"S"ABC                                                                                            3224
CL"S"A                                                                                              3225
DS"Z"AC                                                                                             3226
CL"S"ABC                                                                                            3227
CL"Z                                                                                                3228
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                                                   3229
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++                                    3230
```

```
CL"Z                                                                    3231
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                       3232
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++        3233
EQ"ABCC"AC"AC"£(CL,Q,££(CL,Z),ABC)                                     3234
CL"Z                                                                    3235
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                       3236
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++        3237
CL"Q"AC"ABC                                                             3238
DS"S"AC                                                                 3239
SS"S"ABC                                                                3240
+++++NO SEGMENTATION OF THAT FORM HAS OCCURED+++++                      3241
+++++SS"S"ABC                                                           3242
CL"S"A                                                                  3243
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++        3244
DS"Z"AC                                                                 3245
CL"S"ABC                                                                3246
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++        3247
CL"Z                                                                    3248
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                       3249
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++        3250
CL"Z                                                                    3251
+++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++                       3252
+++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNORED+++++        3253
EQ"AC"AC"AC"£(CL,Q,££(CL,Z),ABC)                                       3254
PS"AC                                                                   3255
AC                                                                      3256
PS"                                                                     3257
+++++THIS PS FUNCTION HAS NULL STRING+++++                              3258
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                           3259
RS                                                                      3260
PS"-------------------------------------------------------------------- 3261
----------------------------------------------------------------------  3262
PS"                                                                     3263
+++++THIS PS FUNCTION HAS NULL STRING+++++                              3264
+++++TO END TRAC JOB ,TYPE ' IN COLUMN 1+++++                           3265
RS                                                                      3266
+++++TRAC JOB FINISHED THANK YOU GOOD-BYE,YEIA-XARA COSTAS+++++         3267
                                                                        3268
```

A P P E N D I X

TRAC INTERPRETER

```
C*******************************************************************************
      BLOCK  DATA
C*******************************************************************************
      IMPLICIT  INTEGER*2  (A-Z)
      COMMON  /STORE/ CHAR,VAL,FSTORE,LINE,NUMBER
      INTEGER*2  CHAR(1000)/1000*' '/,VAL(80)/80*' '/,
     1FSTORE(1000)/Z7FF5,999*' '/,LINE(80)/80*' '/,NUMBER(10)/'0','1','2
     2','3','4','5','6','7','8','9'/
      COMMON/LETTER/A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z
      INTEGER*2A/'A'/,B/'B'/,C/'C'/,D/'D'/,E/'E'/,F/'F'/,G/'G'/,H/'H'/,I
     1/'I'/,J/'J'/,K/'K'/,L/'L'/,M/'M'/,N/'N'/,O/'O'/,P/'P'/,Q/'Q'/,R/'
     2R'/,S/'S'/,T/'T'/,U/'U'/,V/'V'/,W/'W'/,X/'X'/,Y/'Y'/,Z/'Z'/
      COMMON/FUNCT/FA,FB,FC
      INTEGER*2    FA(11) /'R','P','C','D','S','T','T','D','L','P','E'/,
     1             FB(11) /'S','S','L','S','S','N','F','A','N','F','Q'/,
     2             FC(11) /' ','N',' ','N','N','N','N','N','N','N',' '/
      COMMON /PARAM/PARENL,PARENR,COMMA,SHARP,BLANK ,QUOTE,AMPERS
      INTEGER*2   PARENL/'('/,PARENR/')'/,COMMA/','/,SHARP/'£'/,BLANK
     1/' '/ ,QUOTE /''''/ ,AMPERS/'&'/
      COMMON/NEUPOI/PTACT,PTNEU,ENDP,L1,SF,FULL
      INTEGER*2   PTACT/1001/,PTNEU/1001/ ,ENDP/0/,L1/0/,SF/0/,FULL/0/
      COMMON/CONST/BEGP,SPT,PI,META,MEMPTY,FORGET,MARCOM,TOTAL,TRACE
     1,MRQUOT,CARD
      INTEGER*2  BEGP/0/,SPT/1/,PI/1000/ ,META/0/,MEMPTY/Z7FF5/,
     1FORGET/Z7FF6/,MARCOM/Z7FF7/,TOTAL/0/,TRACE/0/,MRQUOT/Z7FF8/,CARD/0
     2/
      COMMON /INPOUT/ READER,PRNTER,IDLE,TIMES
      INTEGER*4  READER/5/,PRNTER/6/,IDLE*2(12)/'£','(','P','S',',',',','£',
     1'(','R','S',')',')',Z7FF8/,TIMES/0/
      END
C*******************************************************************************
C     SUBROUTINE  PROCES
C*******************************************************************************
      IMPLICIT  INTEGER*2  (A-Z)
      COMMON /PARAM/PARENL,PARENR,COMMA,SHARP,BLANK ,QUOTE,AMPERS
      COMMON/CONST/BEGP,SPT,PI,META,MEMPTY,FORGET,MARCOM,TOTAL,TRACE
     1,MRQUOT,CARD
      COMMON /STORE/ CHAR(1000),VAL(80),FSTORE(1000),LINE(80),NUMBER(10)
```

```
      COMMON/NEUPOI/PTACT,PTNEU,ENDP,L1,SF,FULL                          39
      COMMON /FUNCT/FA(11),FB(11),FC(11)                                 40
      DIMENSION  NEU(1000)                                               41
      EQUIVALENCE  (CHAR,NEU)                                            42
      COMMON /INPOUT/ READER,PRNTER,IDLE,TIMES                           43
      INTEGER*4  READER,PRNTER,IDLE*2(12),TIMES*2                        44
      COMMON/LETTER/A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z  45
      DATA NONAME/0/,PTSTOR/1/,FF1/0/,FF2/0/,NULL/'N'/,FORA1/0/,N1/11/   46
      DATA NOSTRN/0/,DIM/ 500/,NEVER/0/,FIRST/1/,AR/1/,N5/1/,SW5/0/      47
      DIMENSION ARREM(5)                                                 48
      DIMENSION  NOCHA(4)                                                49
      DATA  NOCOM/1/,NOCHA/4*0/,METSS/0/,NOFORG/0/                       50
C THE FIRST CHARACTER OF THE ACTIVE STRING IS LOADED WITH THE MARKER END  51
C STRING FOR LOADING THE IDLING PROCEDURE AT THE BEGINNING OF EACH CYCLE  52
      CHAR(1)=MRQUOT                                                     53
 10   IF(CHAR(SPT).EQ.MRQUOT) GO TO 3200                                 54
      IF(CHAR(SPT).EQ.PARENL)  GO TO  20                                 55
      GO TO  80                                                          56
C     SUBROUTINE BEGPAR                                                  57
 20   IF (CHAR(SPT).NE.PARENL)  GO  TO  30                               58
      BEGP=BEGP+1                                                        59
      IF (BEGP.NE.1)  GO  TO  40                                         60
      GO TO 50                                                           61
 30   IF (CHAR(SPT).NE.PARENR)  GO  TO   40                              62
      BEGP=BEGP-1                                                        63
      IF (BEGP.EQ.0) GO TO 70                                            64
 40   NEU(PI)=CHAR(SPT)                                                  65
      PI=PI-1                                                            66
      IF(META.EQ.PI) CALL CLEARP                                         67
      IF(FULL.EQ.1) GO TO 3270                                           68
 50   SPT=SPT+1                                                          69
 60   GO TO 20                                                           70
 70   SPT=SPT+1                                                          71
      GO TO 10                                                           72
 80   IF(CHAR(SPT).EQ.COMMA)  GO TO 150                                  73
 90   IF (CHAR(SPT).EQ.SHARP)  GO  TO  160                               74
 100  IF(CHAR(SPT).EQ.PARENR)  GO  TO  220                               75
 110  NEU(PI)=CHAR(SPT)                                                  76
```

```
 120  PI=PI-1                                                          77
      IF(META.EQ.PI) CALL CLEARP                                       78
      IF(FULL.EQ.1) GO TO 3270                                         79
 130  SPT=SPT+1                                                        80
 140  GO  TO   10                                                      81
C     SUBROUTINE  SUBCOM                                               82
C IT IS EXECUTED WHENEVER A COMMA OCCURS DURING SCANNING               83
150   NEU(PI)=MARCOM                                                   84
      SPT=SPT+1                                                        85
      PI=PI-1                                                          86
      IF(META.EQ.PI) CALL CLEARP                                       87
      IF(FULL.EQ.1) GO TO 3270                                         88
      GO  TO  10                                                       89
C     SUBROUTINE SUBSHA                                                90
 160  ENDP=ENDP+1                                                      91
C      COUNT ACTIVE OR NEUTRAL FUNCTIONS,FOR FINDING THEIR MATCHING  PAR.  92
      SPT=SPT+1                                                        93
      IF (CHAR(SPT).EQ.PARENL)   GO   TO   170                         94
      IF (CHAR(SPT).EQ.SHARP)   GO   TO   180                          95
      GO TO 190                                                        96
 170  SPT=SPT+1                                                        97
      SF=1                                                             98
      NEU(PI)= PTACT                                                   99
      PTACT=PI                                                        100
      PI=PI-1                                                         101
      IF(META.EQ.PI) CALL CLEARP                                      102
      IF(FULL.EQ.1) GO TO 3270                                        103
       GO  TO   10                                                    104
 180  SPT=SPT+1                                                       105
      IF   (CHAR(SPT).EQ.PARENL)   GO    TO    200                    106
      SPT=SPT-1                                                       107
 190  NEU(PI)=SHARP                                                   108
      ENDP=ENDP-1                                                     109
      PI=PI-1                                                         110
      IF(META.EQ.PI) CALL CLEARP                                      111
      IF(FULL.EQ.1) GO TO 3270                                        112
      GO TO 10                                                        113
 200  SPT=SPT+1                                                       114
```

```
      SF=C                                                              115
       NEU(PI)=PTNEU                                                    116
      PTNEU=PI                                                          117
      PI=PI-1                                                           118
      IF(META.EQ.PI) CALL CLEARP                                        119
      IF(FULL.EQ.1) GO TO 3270                                          120
  210 GO  TO  10                                                        121
C     SUBROUTINE  SUBEND                                                122
C THIS IS A SUBR. FOR THE PARENTHESIS MATCHING A FUNCTION               123
  220 SPT=SPT+1                                                         124
  230 IF (ENDP.EQ.0)  GO TO   110                                       125
      ENDP=ENDP-1                                                       126
      IF(TRACE.EQ.1)GO TO 3230                                          127
  240 IF (PTACT.LT.PTNEU)  GO  TO  340                                  128
C THIS IS A NEUTRAL FUNCTION                                            129
      X1= NEU(PTNEU-1)                                                  130
      Y1=NEU(PTNEU-2)                                                   131
  260 DNULL1=BLANK                                                      132
C     SUBROUTINE  LOOKF(X,Y,N,I,DNULL)                                  133
C N1 IS THE NUMBER OF IMPLEMENTED FUNCTIONS                             134
  270 DO   290 I1=1,N1                                                  135
      IF (FA(I1).NE.X1) GO TO 290                                       136
      IF(FB(I1).NE.Y1) GO TO  290                                       137
      IF (FC(I1).EQ.NULL) GO TO 280                                     138
      GO TO 300                                                         139
  280 DNULL1=NULL                                                       140
      GO TO 300                                                         141
  290 CONTINUE                                                          142
      I1=N1+1                                                           143
  300 IF  (PTACT.LT.PTNEU)  GO  TO  360                                 144
      IF(I1.EQ.1)  GO TO 320                                            145
  310 GO  TO  380                                                       146
C DELETE NEUTRAL FUNCTION FROM NEUTRAL STRING                           147
  320 PI=PTNEU                                                          148
      PTNEU=NEU(PI)                                                     149
  330 GO TO 380                                                         150
C  THIS  IS  AN  ACTIVE  FUNCTION                                       151
  340 X1=NEU(PTACT-1)                                                   152
```

```
      Y1=NEU(PTACT-2)                                                    153
C  SF=1  MEANS ACTIVE   FUNCTION                                         154
 350   SF=1                                                             155
      GC TC 260                                                         156
 360  IF(I1.EQ.1)  GC TC 370                                            157
      GC TO 380                                                         158
C DELETE ACTIVE FUNCTICN FRCM NEUTRAL STRING                            159
 370  PI=PTACT                                                          160
      PTACT=NEU(PI)                                                     161
 380   GO TO (390,680,810,1420,1870,2290,2300,2310,2320,2480,2815,3100) 162
     1,I1                                                               163
C*************************************************************************** 164
C     SUBROUTINE   RS                                                   165
C IT READS CHARACTER STRINGS FROM THE TYPEWRITER                        166
C READS A CARD AT A TIME                                                167
 390  CARD=0                                                            168
 400  READ(READER,410,END=630)  (VAL(J),J=1,80)                         169
 410  FORMAT (80A1)                                                     170
      CARD=CARD+1                                                       171
C650  FORMAT (' CARD=',I3,5X,80A1)                                      172
C65   WRITE (PRNTER,650)CARD,  (VAL(J),J=1,80)                          173
C IF FIRST READ CHARACTER IS ' JCB FINISHES                             174
      IF (VAL(1).EQ.QUOTE.AND.CARD .EQ.1)  GO  TO  670                  175
      L1=1                                                              176
 420  IF (VAL(L1).EQ.QUCTE)  GC TC 430                                  177
      IF  (L1.EQ.80)  GO  TO  440                                       178
      L1=L1+1                                                           179
      GO  TO  420                                                       180
 430  S1=QUCTE                                                          181
 440  IF (SF.EQ.1)  GO  TO  480                                         182
C     SUBROUTINE   NSTRNG                                               183
C FOR PUTING IN NEUTRAL STRING THE VALUE OF A NEUTRAL FUNCTION          184
      DC 450  K=1,80                                                    185
      IF (VAL(K).EQ.QUCTE) GC TC 460                                    186
      IF(META.EQ.PI) CALL CLEARP                                        187
      IF(FULL.EQ.1) GC TC 670                                           188
      NEU(PI)=VAL(K)                                                    189
 450  PI=PI-1                                                           190
```

```
 460  IF (S1.NE.QUOTE)  GO  TO  400                                    191
      S1=BLANK                                                         192
      L1=0                                                             193
      GO TO 670                                                        194
C     SUBROUTINE  ASTRNG                                               195
C FOR PUTING IN THE ACTIVE STRING THE VALUE OF AN ACTIVE FUNCTION      196
 480  TELOS=VAL(L1)                                                    197
      DIF=META-SPT+1                                                   198
C 1 ADDED TO INCLUDE THE ENDING CHARACTER ' TO  THE MOVE OF UNSCANNED CH 199
C  L1=NUMBER OF CHARACTERS OF THE VALUE OF THE ACTIVE FUNCTION          200
      IF(VAL(L1).EQ.QUOTE)  GO  TO  610                                201
 490  TDIF=DIF                                                         202
      NL1=L1+TIMES*80                                                  203
      IF ((NL1+DIF).LT.META)  GO  TO  580                              204
      IF ((NL1+DIF).GE.PI) GO TO 650                                   205
 500  CHAR(NL1+DIF )=CHAR(META)                                        206
 510  META=META-1                                                      207
      DIF=DIF-1                                                        208
      IF (META.GE.SPT)  GO  TO  500                                    209
 520  META=NL1+TDIF                                                    210
      SPT=1+TIMES*80                                                   211
C ALL VALUES  FINISH  WITH QUOTE '                                     212
 530  DO  550  IND=1,L1                                                213
 540  CHAR(SPT)=VAL(IND)                                               214
 550  SPT=SPT+1                                                        215
      IF (TELOS.EQ.QUOTE)  GO  TO  560                                 216
      TIMES=TIMES+1                                                    217
      GO TO 620                                                        218
 560  TIMES=0                                                          219
      SPT=1                                                            220
 570  GO TO 620                                                        221
C IN THAT CASE WE MOVE FIRST THE CHARACTER WHITH THE LOWEST SUBSCRIPT   222
 580  DIF1=1                                                           223
 590  CHAR(NL1+DIF1)=CHAR(SPT)                                         224
 600  DIF1=DIF1+1                                                      225
      IF (DIF1.GT.DIF)  GO  TO  520                                    226
      SPT=SPT+1                                                        227
      GO TO 590                                                        228
```

```
 610  L1=L1-1                                                                229
      IF(L1.NE.0) GO TO 490                                                  230
      VAL(1)=BLANK                                                           231
      GO TO 560                                                              232
 620  GO  TO  460                                                            233
 630  WRITE(PRNTER,640)                                                      234
 640  FORMAT (' +++++END OF DATA  WHITHOUT TERMINATING ''++++++')            235
      GO TO 670                                                              236
 650  WRITE(PRNTER,660)                                                      237
 660  FORMAT(' +++++VALUE OF ACTIVE RS EXCEEDS LIMITS PROCESSOR FULL++++    238
     1+')                                                                    239
      FULL=1                                                                 240
C WHEN FIRST READ CHARACTER IS EQUAL ' THE TRAC JOB FINISHES                 241
 670  IF (VAL(1).EQ.QUOTE.AND.CARD .EQ.1)  GO  TO  3170                      242
      IF (FULL.EQ.1) GO TO 3270                                              243
      GO   TO   3160                                                         244
C*************************************************************************  245
C     SUBROUTINE PS                                                          246
 680  IF (SF.NE.1       )   GO  TO  740                                      247
C   THIS  IS  ACTIVE   FUNCTION  NEXTCH SHOULD BE A COMMA                    248
      IF (NEU(PTACT-3).NE.MARCOM)  GO  TO  760                               249
      LOC2=PTACT-4                                                           250
 690  IF (PI.EQ.LOC2)  GO TO 720                                             251
 700  LENGTH=LOC2-PI                                                         252
      WRITE (PRNTER,710)(NEU(LOC2+1-J2),J2=1,LENGTH)                         253
 710  FORMAT(' ',79A1)                                                       254
      GO  TO  790                                                            255
 720  WRITE(PRNTER,730)                                                      256
 730  FORMAT(' +++++THIS PS FUNCTION HAS NULL STRING+++++')                  257
      GO  TO  790                                                            258
C  THIS  IS  NEUTRAL FUNCTION NEXTCH SHOULD BE A COMMA                      259
 740  IF  (NEU(PTNEU-3).NE.MARCOM)  GO  TO 750                               260
      LOC2=PTNEU-4                                                           261
      GO TO 690                                                              262
 750  LOC2=PTNEU                                                             263
      GO  TO  770                                                            264
 760  LOC2=PTACT                                                             265
 770  LENGTH=LOC2-PI-1                                                       266
```

```
      WRITE (PRNTER,780) (NEU(LOC2-J2),J2=1,LENGTH)                    267
  780 FORMAT(' +++++ERROR FUNCTION*****',55A1,'+++++')                 268
  790 IF(SF.EQ.1)  GO  TO  800                                         269
C DELETE NEUTRAL FUNCTION FROM NEUTRAL STRING                          270
      PI=PTNEU                                                         271
      PTNEU=NEU(PI)                                                    272
      GO   TO   3160                                                   273
C DELETE ACTIVE FUNCTION FROM NEUTRAL STRING                           274
  800 PI=PTACT                                                         275
      PTACT=NEU(PI)                                                    276
      GO  TO  3160                                                     277
C*************************************************************************278
C     SUBROUTINE CL                                                    279
C THIS SUBROUTINE BRINGS BACK THE VALUE OF A FORM                      280
  810 IF (SF.EQ. 1)  GO  TO  820                                       281
      KOINC=PTNEU                                                      282
       GO TO 830                                                       283
  820 KOINC=PTACT                                                      284
  830 LOC4=KOINC-3                                                     285
      IF (NEU(LOC4).NE.MARCOM)  GO  TO  850                            286
  840 LOC4=LOC4-1                                                      287
      IF  (NEU(LOC4).EQ.MARCOM)  GO  TO  900                           288
      IF (LOC4.LE.PI)  GO  TO  940                                     289
      NONAME=NONAME+1                                                  290
      GO  TO  840                                                      291
  850 WRITE(PRNTER,860)                                                292
  860 FORMAT  (' +++++ERROR FUNCTION ,COMMA IS MISSING AFTER FIRST ARGUM 293
     1ENT+++++')                                                       294
  870 LENGTH=KOINC-PI-1                                                295
      WRITE (PRNTER,880) (NEU(KOINC-M1),M1=1,LENGTH)                   296
  880 FORMAT(' +++++',69A1,'+++++')                                    297
      IF (SF.EQ.1)  GO  TO  890                                        298
C DELETE NEUTRAL FUNCTION FROM NEUTRAL STRING                          299
      PI=PTNEU                                                         300
      PTNEU=NEU(PI)                                                    301
      GO  TO  1180                                                     302
  890 PI=PTACT                                                         303
C DELETE ACTIVE FUNCTION FROM NEUTRAL STRING                           304
```

```
      PTACT=NEU(PI)                                                    305
      GO  TO  1180                                                     306
C SEARCH FOR MATCHING OF THE NAME                                      307
 900  IF ((LOC4-1).LE.PI)  GO  TO  940                                 308
 910  IF  (FSTORE(PTSTOR).EQ.MEMPTY)  GO  TO  960                      309
 920  IF  (NONAME.EQ.FSTORE(PTSTOR+1))  GO  TO  980                    310
 930  PTSTOR=FSTORE(PTSTOR)+PTSTOR                                     311
      GO  TO  910                                                      312
 940  WRITE (PRNTER,950)                                               313
 950  FORMAT(' +++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++')     314
      FF1=1                                                            315
      GO  TO  910                                                      316
 960  WRITE  (PRNTER,970)                                              317
 970  FORMAT (' +++++THERE IS NO SUCH FORM IN FSTORE TO BE CALLED   ++++ 318
     1+')                                                              319
      GO  TO  870                                                      320
 980  IF (NONAME.EQ.0)  GO  TO  1010                                   321
 990  PTSTR1=PTSTOR+1                                                  322
      KOINC3=KOINC-3                                                   323
      DO 1000  N=1,NONAME                                              324
      IF (FSTORE(PTSTR1+N).NE.NEU(KOINC3-N))  GO  TO  930              325
1000  CONTINUE                                                         326
C AFTER  MATCH OF NAME SEARCH FOR MATCHING ARGUMENTS                   327
1010 FO=PTSTOR+NONAME+3                                                328
      F2=FO-1                                                          329
      FTEL=FSTORE(PTSTOR)+PTSTOR                                       330
      IF ((FO+1).EQ.FTEL) GO TO 1390                                   331
      NOARG=1                                                          332
1020 IF  (FSTORE(F2).EQ.0)  GO TO 1230                                 333
      N5=FSTORE(F2)                                                    334
      L3=FO+1                                                          335
      PROS=TOTAL+1                                                     336
      LTOTAL=PROS                                                      337
C  AFTER MATCH OF NAME SEARCH FOR MARKERS                              338
1030 DO  1040  M5=1,N5                                                 339
      TM5=M5                                                           340
      MARKER=MEMPTY-M5                                                 341
      IF (FSTORE(L3).EQ.MARKER)  GO TO 1250                            342
```

```
 1040 CONTINUE                                                          343
C THE EXAMINED CHARACTER WAS NOT A MARKER ,IT IS MOVED AT THE END OF FST 344
      FSTORE(PROS)=FSTORE(L3)                                           345
 1050 L3=L3+1                                                           346
      IF  (L3.LT.FTEL)  GO  TO 1070                                     347
 1060 FF1=0                                                             348
      GO TO 1080                                                        349
 1070 PROS=PROS+1                                                       350
      IF (PROS.GT.DIM) GO TO 1370                                       351
      GO  TO  1030                                                      352
C THE SUBSTITUTION OF MARKERS FINISHES                                  353
 1080 L1=PROS-LTOTAL+1                                                  354
 1090 IF  (SF.EQ.1) GO TO 1120                                          355
C DELETE NEUTRAL FUNCTION FROM NEUTRAL STRING                           356
      PI=PTNEU                                                          357
      PTNEU=NEU(PI)                                                     358
      IT=PI                                                             359
C THE VALUE OF THE NEUTRAL FUNCTION GOES ON TOP OF NEUTRAL STRING       360
      DO  1100 K=LTOTAL,PROS                                            361
      IF(META.EQ.PI) CALL CLEARP                                        362
      IF(FULL.EQ.1) GO TO 1180                                          363
      NEU(PI)=FSTORE(K)                                                 364
 1100 PI=PI-1                                                           365
 1110 GO TO 1170                                                        366
C  MOVE ACTIVE STACK LEFT OR RIGHT TO PUT THE VALUE OF ACTIVE FUNCTION  367
C DELETE ACTIVE FUNCTION FROM NEUTRAL STRING                            368
 1120 PI=PTACT                                                          369
      PTACT=NEU(PI)                                                     370
      DIF=META-SPT+1                                                    371
      TDIF=DIF                                                          372
C 1 ADDED TO INCLUDE THE ENDING CHARACTER ' TO  THE MOVE OF UNSCANNED CH 373
C  L1=NUMBER OF CHARACTERS OF THE VALUE OF THE ACTIVE FUNCTION          374
      IF ((L1+DIF).EQ.META)   GO  TO  1150                              375
      IF ((L1+DIF).LT.META)   GO  TO  1200                              376
      IF ((L1+DIF).GE.PI)  GO TO 1350                                   377
 1130 CHAR(L1+DIF )=CHAR(META)                                          378
 1140 META=META-1                                                       379
      DIF=DIF-1                                                         380
```

```
      IF (META.GE.SPT)  GO  TO  1130                                    381
 1150 META=L1+TDIF                                                      382
      SPT=1                                                             383
      DO  1160 IND=LTOTAL,PROS                                          384
      CHAR(SPT)=FSTORE(IND)                                             385
 1160 SPT=SPT+1                                                         386
      SPT=1                                                             387
 1170 IF (FF2.EQ.1)  GO TO 1340                                         388
CPUT END MARKER AFTER LAST CHARACTER OF FSTORE FSTORE HAS NOW SO MANY CH 389
C ACTERS AS BEFORE THE FUNCTION CALL                                    390
      FSTORE(LTOTAL)=MEMPTY                                             391
 1180 N5=1                                                              392
      PTSTOR=1                                                          393
      DEIKTH=1                                                          394
      NONAME=0                                                          395
 1190 FF1=0                                                             396
      GO TO 1410                                                        397
C IN THAT CASE WE MOVE FIRST THE CHARACTER WHITH THE LOWEST SUBSCRIPT    398
 1200 DIF1=1                                                            399
 1210 CHAR(L1+DIF1)=CHAR(SPT)                                           400
 1220 DIF1=DIF1+1                                                       401
      IF (DIF1.GT.DIF)  GO  TO  1150                                    402
      SPT=SPT+1                                                         403
      GO TO 1210                                                        404
 1230 WRITE (PRNTER,1240)                                               405
 1240 FORMAT(' +++++FORM WITH NO MARKERS ,ARGUMENTS OF CL FUNCTION IGNOR 406
     1ED+++++')                                                         407
      LTOTAL=FO+1                                                       408
      PROS=FTEL-1                                                       409
C THE FORM IS NOT MOVED AT THE END OF FSTORE,ITS VALUE IS TAKEN FROM ITS 410
      FF2=1                                                             411
      GO  TO  1080                                                      412
 1250 IF  (FF1.NE.1)  GO TO 1280                                        413
      WRITE (PRNTER,1260)                                               414
 1260 FORMAT(' +++++THE CL FUNC HAS NO ARGUMENTS MARKERS REPLACED BY NUL 415
     1L+++++')                                                          416
      IF((L3+1).GE.FTEL) GO TO 1270                                     417
      L3=L3+1                                                           418
```

```
      GO  TO  1030                                                          419
 1270 PROS=PROS-1                                                           420
      GO TO 1060                                                            421
 1280 IF  (NOARG.EQ.TM5)  GO TO 1320                                        422
 1290 LOC4=LOC4-1                                                           423
      IF  (LOC4.LE.PI)  GO  TO  1300                                        424
      IF (NEU(LOC4).NE.MARCOM)  GO  TO  1290                                425
      IF (NEU(LOC4-1).EQ.MARCOM) GO TO  1290                                426
      NOARG=NOARG+1                                                         427
      GO  TO 1280                                                           428
 1300 WRITE (PRNTER,1310)                                                   429
 1310 FORMAT (' +++++NUMBER OF ARGUMENTS LESS THAN NO OF MARKERS'/          430
     1' NOMARKERS-NOARG MARKERS REPLACED BY NULL+++++')                     431
      GO  TO   1330                                                         432
C BEGIN SUBSTITUTION OF THAT ARGUMENT IN FSTORE UNTIL COMMA OCCURS          433
 1320 LOC4=LOC4-1                                                           434
      IF  (LOC4.LE.PI)  GO  TO 1330                                         435
      IF (NEU(LOC4).EQ.MARCOM) GO TO 1330                                   436
      FSTORE(PROS)=NEU(LOC4)                                                437
      PROS=PROS+1                                                           438
      IF (PROS.GT.DIM) GO TO 1370                                           439
      GO TO 1320                                                            440
 1330 NOARG=1                                                               441
      LOC4=KOINC-NONAME-4                                                   442
      PROS=PROS-1                                                           443
      GO  TO  1050                                                          444
 1340 FF2=0                                                                 445
      GO TO 1180                                                            446
 1350 WRITE (PRNTER,1360)                                                   447
 1360 FORMAT(' +++++VALUE OF ACTIVE CL EXCEEDS LIMITS PROCESSOR FULL++++    448
     1+')                                                                   449
      FULL=1                                                                450
      GO TO 1180                                                            451
 1370 WRITE (PRNTER,1380)                                                   452
 1380 FORMAT(' +++++NO PLACE IN FSTORE FORM CANNOT BE CALLED+++++')         453
      L1=1                                                                  454
C NUMBER OF VALUE CHARACTERS IS SET TO 1 TO AVOID CONFUSION WITH NULL FO    455
      GO TO 870                                                             456
```

```
 1390 WRITE(PRNTER,1400)                                                    457
 1400 FORMAT(' +++++CALL TO A FORM WITH NULL VALUE+++++')                   458
      GO TO 870                                                             459
 1410 IF (FULL.EQ.1) GO TO 3270                                             460
      GO TO 3160                                                            461
C*************************************************************************  462
C     SUBROUTINE DS                                                        463
 1420 IF (FSTORE(1).EQ.MEMPTY) NEVER=0                                      464
C FSTORE WAS FULL AND ALL FORMS WERE DELRTED WITH DA FUNCTION               465
      IF (NEVER.EQ.1)  GO  TO  1650                                         466
 1430 IF (SF.EQ.1)  GO  TO  1440                                            467
      KOINC=PTNEU                                                           468
      GO  TO  1450                                                          469
 1440 KOINC=PTACT                                                           470
 1450 LOC4=KOINC-3                                                          471
      IF (NEU(LOC4).NE.MARCOM)  GO  TO  1480                                472
 1460 LOC4=LOC4-1                                                           473
      IF (NEU(LOC4).EQ.MARCOM)  GO  TO 1610                                 474
      IF (LOC4.LE.PI) GO TO 1470                                            475
      NONAME=NONAME+1                                                       476
      GO  TO  1460                                                          477
C ONE IS ADDED SINCE IT IS SUBTRACTED AGAIN WHEN THE FORM IS CREATED         478
 1470 LOC4=LOC4+1                                                           479
      GO TO 1610                                                            480
 1480 WRITE  (PRNTER,1490)                                                  481
 1490 FORMAT( ' +++++ERROR FUNCTION  THERE IS NO COMMA AFTER FIRST ARGUM    482
     1ENT+++++')                                                            483
 1500 LENGTH=KOINC-PI-1                                                     484
      WRITE (PRNTER,1600) (NEU(KOINC-M1),M1=1,LENGTH)                       485
 1600 FORMAT(' +++++UNDEFINED STRING FOR THE FOLLOWING FUNCTION*****',/     486
     1' +++++',69A1,'+++++')                                                487
      GO TO 790                                                             488
C SEARCH FOR EMPTY STORE                                                    489
 1610 IF (FSTORE(PTSTOR).NE.MEMPTY)  GO TO 1810                             490
 1620 DEIKTH=PTSTOR                                                         491
      PTSTOR=PTSTOR+1                                                       492
 1630 IF (PTSTOR.LE.DIM) GO TO 1670                                         493
 1640 NEVER=1                                                               494
```

```
 1650 LENGTH=KOINC-PI-1                                                  495
      WRITE (PRNTER,1660) (NEU(KOINC-M1),M1=1,LENGTH)                    496
 1660 FORMAT(' +++++THERE IS NO PLACE IN STORE AVAILABLE FOR FORMS+++++'  497
     1,/' +++++',69A1,'+++++')                                           498
      GO TO 1770                                                         499
 1670 FSTORE(PTSTOR)=NONAME                                              500
      IF (NONAME.EQ.0)  GO  TO  1690                                     501
C PUT THE NAME OF STRING IN STORE                                        502
      DO 1680  N=1,NONAME                                                503
      PTSTOR=PTSTOR+1                                                    504
      IF(PTSTOR.LE.DIM) GO TO 1680                                       505
      GO TO 1640                                                         506
 1680 FSTORE(PTSTOR)=NEU(KOINC-3-N)                                      507
C WHEN NON SEGMENT STRING ONE PLACE AFTER NAME PUT 0                     508
 1690 PTSTOR=PTSTOR+1                                                    509
      IF (PTSTOR.LE.DIM) GO TO 1700                                      510
      GO TO 1640                                                         511
 1700 FSTORE(PTSTOR)=0                                                   512
C THE FORM POINTER TWO PLACES AFTER NAME POINTS NEXT CHARACTER OF THE FO 513
C HERE IS 1 TO POINT FIRST CHARACTER OF THE FORM                         514
      PTSTOR=PTSTOR+1                                                    515
      IF (PTSTOR.LE.DIM) GO TO 1710                                      516
      GO TO 1640                                                         517
 1710 FSTORE(PTSTOR)=1                                                   518
C THE ACTUAL STRING IS NOW PASSED TO THE FORM                            519
 1720 LOC4=LOC4-1                                                        520
      IF(LOC4.LE.PI) GO TO 1740                                          521
      PTSTOR=PTSTOR+1                                                    522
      IF (PTSTOR.LE.DIM) GO TO 1730                                      523
      GO TO 1640                                                         524
 1730 NOSTRN=NOSTRN+1                                                    525
      FSTORE(PTSTOR)=NEU(LOC4)                                           526
      GO TO 1720                                                         527
 1740 IF  (NOSTRN.EQ.0) GO TO 1790                                       528
 1750 PTSTOR=PTSTOR+1                                                    529
      IF (PTSTOR.LE.DIM) GO TO 1760                                      530
      GO TO 1640                                                         531
 1760 FSTORE(DEIKTH)=PTSTOR-DEIKTH                                       532
```

```
      TOTAL=TOTAL+FSTORE(DEIKTH)                                        533
      FSTORE(PTSTOR)=MEMPTY                                             534
 1770  PTSTOR=1                                                         535
      NONAME=0                                                          536
      NOSTRN=0                                                          537
      DEIKTH=1                                                          538
 1780 GO TO 790                                                         539
 1790 LENGTH=KOINO-PI-1                                                 540
      WRITE (PRNTER,1800) (NEU(KOINO-M1),M1=1,LENGTH)                   541
 1800 FORMAT(' +++++THIS STRING HAS NULL VALUE+++++',/' +++++',69A1,'+++ 542
     1++')                                                              543
      GO TO 1750                                                        544
C IN NON EMPTY STORE SEARCH FOR MATCH OF NAME                           545
 1810 DEIKTH=PTSTOR                                                     546
      IF (NONAME.EQ.FSTORE(PTSTOR+1)) GO TO 1830                        547
 1820 PTSTOR=FSTORE(PTSTOR)+DEIKTH                                      548
      GO TO 1610                                                        549
 1830 IF (NONAME.EQ.0)  GO TO 1850                                      550
      DO 1840 N=1,NONAME                                                551
      IF(FSTORE(PTSTOR+1+N).NE.NEU(KOINO-3-N)) GO TO 1820               552
 1840 CONTINUE                                                          553
C SO  MANY CHARACTERS AS THEY ARE IN PLACE OF THE DELETED FORM          554
 1850 SHIFT=FSTORE(PTSTOR)                                              555
      MARX=PTSTOR+SHIFT                                                 556
      MTEL=TOTAL+1                                                      557
      DO  1860 M=MARX,MTEL                                              558
 1860 FSTORE(M-SHIFT)=FSTORE(M)                                         559
      PTSTOR=MTEL-SHIFT                                                 560
      TOTAL=TOTAL-SHIFT                                                 561
      GO TO 1620                                                        562
C*************************************************************************563
C     SUBROUTINE   SS                                                   564
 1870 IF (SF.EQ. 1)  GO  TO  1880                                       565
      KOINO=PTNEU                                                       566
       GO TO 1890                                                       567
 1880 KOINO=PTACT                                                       568
 1890 LOC4=KOINO-3                                                      569
      IF (NEU(LOC4).NE.MARCOM)  GO  TO  1910                            570
```

```
 1900 LOC4=LOC4-1                                                        571
      IF  (NEU(LOC4).EQ.MARCOM)  GO  TO  1950                            572
      IF (LOC4.LE.PI)  GO  TO  1980                                      573
      NONAME=NONAME+1                                                    574
      GO  TO  1900                                                       575
 1910 WRITE(PRNTER,1920)                                                 576
 1920 FORMAT  (' +++++ERROR FUNCTION ,COMMA IS MISSING AFTER FIRST ARGUM 577
     1ENT+++++'/' +++++STRING NON SEGMENTED+++++')                       578
 1930 LENGTH=KOINC-PI-1                                                  579
      WRITE (PRNTER,1940) (NEU(KOINC-M1),M1=1,LENGTH)                    580
 1940 FORMAT(' +++++',69A1,'+++++')                                      581
      GO TO  2270                                                        582
C SEARCH FOR MATCHING OF THE NAME                                        583
 1950 IF ((LOC4-1).LE.PI)  GO  TO  1980                                  584
 1960 IF  (FSTORE(PTSTOR).EQ.MEMPTY)  GO  TO  2000                       585
       DEIKTH=PTSTOR                                                     586
      IF (NONAME.EQ.FSTORE(PTSTOR+1))  GO TO 2020                        587
 1970 PTSTOR=FSTORE(DEIKTH)+DEIKTH                                       588
      GO  TO  1960                                                       589
 1980  WRITE (PRNTER,1990)                                               590
 1990 FORMAT(' +++++THERE ARE NO ARGUMENTS IN THAT FUNCTION+++++')       591
      GO  TO  1930                                                       592
 2000 WRITE  (PRNTER,2010)                                               593
 2010 FORMAT (' +++++THERE IS NO SUCH FORM IN FSTORE TO BE SEGMENTED++++ 594
     1+')                                                                595
      GO  TO  1930                                                       596
 2020 PTSTR1=PTSTOR+1                                                    597
      KOINC3=KOINC-3                                                     598
      IF (NONAME.EQ.0) GO TO 2040                                        599
C WHEN THERE IS A FORM WITH NULL NAME SEARCH ONLY FOR NONAME=0 IN FSTORE  600
      DO  2030 N=1,NONAME                                                601
      IF (FSTORE(PTSTR1+N).NE.NEU(KOINC3-N))  GO  TO  1970               602
 2030  CONTINUE                                                          603
C AFTER  MATCH OF NAME SEARCH FOR MATCHING ARGUMENTS                     604
 2040 MARKER=MEMPTY-N5                                                   605
      FTEL=FSTORE(PTSTOR)+PTSTOR                                         606
      FO=PTSTOR+NONAME+3                                                 607
      F2=FO-1                                                            608
```

```
2050 FARX=F0                                                          609
     IF (SW5.EQ.1)  GO  TO  2080                                      610
     SW5=1                                                            611
2060 LOC4=LOC4-1                                                      612
     IF (LOC4.LE.PI)  GO  TO  2110                                    613
2070 IF (NEU(LOC4).EQ.MARCOM)  GO  TO  2120                           614
2080 FARX=FARX+1                                                      615
2090 IF  (FARX.GE.FTEL)  GO  TO  2150                                 616
     IF (NEU(LOC4).EQ.FSTORE(FARX))  GO  TO  2100                     617
     IF  (METSS.EQ.0)  GO  TO  2080                                   618
     LOC4=LOC4+METSS                                                  619
     FARX=FARX-METSS+1                                                620
     METSS=0                                                          621
     GO TO 2090                                                       622
2100 METSS=METSS+1                                                    623
     GO  TO  2060                                                     624
2110 IF(FARX.GE.FTEL)  GO  TO  2180                                   625
2120 IF(METSS.EQ.0) GO TO 2160                                        626
     FSTORE(F2)=N5                                                    627
     LOC4=LOC4+METSS                                                  628
     ARX=FARX-METSS+1                                                 629
     FSTORE(ARX)=MARKER                                               630
     ARX=ARX+1                                                        631
     IF (METSS.EQ.1)  GO TO 2140                                      632
     DO  2130  NSEG=ARX,FARX                                          633
2130 FSTORE(NSEG)=FORGET                                              634
2140 METSS=0                                                          635
     GO  TO  2080                                                     636
2150 N5=N5+1                                                          637
     MARKER=MEMPTY-N5                                                 638
     METSS=0                                                          639
2160 SW5=0                                                            640
2170 IF  (NEU(LOC4).EQ.MARCOM)  GO  TO  2050                          641
     LOC4=LOC4-1                                                      642
     IF  (LOC4.LE.PI)  GO  TO  2180                                   643
     GO TO 2170                                                       644
2180 IF (FSTORE(F2).EQ.0) GO  TO 2250                                 645
     LTOTAL=TOTAL                                                     646
```

```
C MOVE THE STRING WHICH IS NCN SEGMENTED, TC THE RIGHT DELETING FORGET CH.
 2190 IF (FSTORE(PTSTOR).EQ.FORGET) GC TO 2200
      NOFORG=NOFORG+1
      TOTAL=LTOTAL+NOFORG
      IF (TOTAL.GT.DIM) GO TO 2210
      FSTORE(TOTAL)=FSTORE(PTSTOR)
 2200 PTSTOR=PTSTOR+1
      IF (PTSTOR.GE.FTEL)  GO  TO  2230
      GC TO 2190
 2210 WRITE (PRNTER,2220)
 2220 FORMAT(' +++++FORM CANNOT BE SEGMENTED,STORE FULL+++++')
      GO TO 1930
C MOVING THE SEGMENTED STRING TO THE RIGHT HAS FINISHED
C THE OLD STRING WITH FORGET CHARACTERS IS DELETED BY MOVING TO THE LEFT
 2230 IF ((TOTAL+1).GT.DIM) GO TO 2210
      FSTORE(TOTAL+1)=MEMPTY
C  THE SHIFTING FOLLOWS
C THE FIRST CHAR OF MOVED FORM MUST CONTAIN THE NUMBER NOFORG IN PLACE 1
      LTOTAL=LTOTAL+1
      FSTORE(LTOTAL)=NOFORG
C FIRST POSITION IN FORM SHOULD CONTAIN THE NUMBER OF NOFORG CHAR
      SHIFT=FSTORE(DEIKTH)
      MTEL=TOTAL+1
      DO 2240 M=FTEL,MTEL
 2240 FSTORE(M-SHIFT)=FSTORE(M)
      TOTAL=TOTAL-SHIFT
      GO TO 2270
 2250 WRITE (PRNTER,2260)
 2260 FORMAT(' +++++NO SEGMENTATION OF THAT FORM HAS OCCURED+++++')
      GO TO 1930
 2270 NONAME=0
      N5=1
      SW5=0
      METSS=0
      NOFORG=0
      PTSTOR=1
      DEIKTH=1
 2280 GO TO 790
```

```
C*************************************************************************      685
C      SUBROUTINE  TN                                                          686
C A FLAG IS SET ON AND EACH FUNCTION IS COPIED FROM THE NEUTRAL STRINGBE        687
C EVALUATION                                                                   688
 2290 TRACE=1                                                                  689
      GO TO 790                                                                690
C*************************************************************************      691
C      SUBROUTINE  TF                                                          692
C THE FLAG IS SET OFF AND EXECUTION OF TRACE STOPS AFTER THAT                  693
 2300 TRACE=0                                                                  694
      GO TO 790                                                                695
C*************************************************************************      696
C      SUBROUTINE  DA                                                          697
 2310 FSTORE(1)=MEMPTY                                                         698
      TOTAL=0                                                                  699
      GO TO 790                                                                700
C*************************************************************************      701
C      SUBROUTINE  LN                                                          702
C LISTING THE NAMES OF FORMS IN STORE WITH A STRING OF CHAR. IN FRONT          703
 2320 PTSTOR=1                                                                 704
      METLN=0                                                                  705
      IF(SF.EQ.1) GO TO 2330                                                   706
      KOINC=PTNEU                                                              707
      GO TO 2340                                                               708
 2330 KOINC=PTACT                                                              709
 2340 LOC4=KOINC-3                                                             710
      IF(NEU(LOC4).NE.MARCOM)  GO TO 2360                                      711
 2350 LOC4=LOC4-1                                                              712
      IF(LOC4.LE.PI) GO TO 2410                                                713
      METLN=METLN+1                                                            714
      GO TO 2350                                                               715
 2360 WRITE(PRNTER,2370)                                                       716
 2370 FORMAT(' +++++ERROR FUNCTION,COMMA IS MISSING AFTER FIRST ARGUMENT       717
     1+++++')                                                                  718
 2380 LENGTH=KOINC-PI-1                                                        719
      WRITE(PRNTER,2390) (NEU(KOINC-KK),KK=1,LENGTH)                           720
 2390 FORMAT(' +++++',69A1,'+++++')                                            721
 2400 NONAME=0                                                                 722
```

```
      PTSTOR=1                                                          723
      GO TO 790                                                         724
C SEARCH THE FSTORE FOR FORMS                                           725
 2410 IF(FSTORE(PTSTOR).EQ.MEMPTY) GO TO 2430                           726
      NONAME=FSTORE(PTSTOR+1)                                           727
      IF(METLN.EQ.0) GO TO 2450                                         728
      IF(NONAME.EQ.0)  GO TO 2460                                       729
C THE SECOND ARGUMENT OF THE LN FUNCTION IS WRITTEN FOLLOWED BY THE NAME OF     730
C EACH FORM THAT EXISTS IN STORE                                        731
      METLN=METLN+3                                                     732
      NONAME=NONAME+2                                                   733
      WRITE(PRNTER,2420)(NEU(KOINO-JJ),JJ=4,METLN),(FSTORE(PTSTOR+KK),KK      734
     1=2,NONAME)                                                        735
 2420 FORMAT(' ',79A1)                                                  736
      METLN=METLN-3                                                     737
      NONAME=NONAME-2                                                   738
      GO TO 2470                                                        739
 2430 WRITE (PRNTER,2440)                                               740
 2440 FORMAT(' +++++THERE ARE NO FORMS IN STORE+++++')                  741
      GO TO 2400                                                        742
 2450 IF(NONAME.EQ.  0) GO TO 2470                                      743
C THE SECOND ARGUM OF LN FUNCTION IS NULL ONLY THE NAME OF THE FORM IS WRIT     744
      NONAME=NONAME+2                                                   745
      WRITE(PRNTER,2420) (FSTORE(PTSTOR+KK),KK=2,NONAME)                746
      NONAME=NONAME-2                                                   747
      GO TO 2470                                                        748
 2460 METLN=METLN+3                                                     749
C THE NAME OF THE FORM IS NULL ONLY THE SECOND ARGUMENT OF LN FUNC.IS WRIT.      750
      WRITE (PRNTER,2420) (NEU(KOINO-JJ),JJ=4,METLN)                    751
      METLN=METLN-3                                                     752
C BOTH NAME OF THE FORM AND SEC.ARG. OF LN FUNC. ARE NULL NOTHING IS WRIT.      753
 2470 PTSTOR=PTSTOR+FSTORE(PTSTOR)                                      754
C SEARCH FSTORE FOR NEXT FORM                                           755
      GO TO 2410                                                        756
C*************************************************************************     757
C     SUBROUTINE  PF                                                    758
 2480 IF (SF.EQ. 1)  GO  TO  2490                                       759
      KOINO=PTNEU                                                       760
```

```
      GO TO 2500                                                        761
 2490 KOINO=PTACT                                                       762
 2500 LOC4=KOINO-3                                                      763
      IF (NEU(LOC4).NE.MARCOM)  GO  TO  2540                            764
 2510 LOC4=LOC4-1                                                       765
      IF (LOC4.LE.PI) GO TO 2520                                        766
      NONAME=NONAME+1                                                   767
      GO TO 2510                                                        768
 2520 IF  (FSTORE(PTSTOR).EQ.MEMPTY)  GO  TO  2590                      769
      IF (NONAME.EQ.FSTORE(PTSTOR+1))  GO TO 2610                       770
 2530 PTSTOR=FSTORE(PTSTOR)+PTSTOR                                      771
      GO TO 2520                                                        772
 2540 WRITE(PRNTER,2550)                                                773
 2550 FORMAT  (' +++++ERROR FUNCTION ,COMMA IS MISSING AFTER FIRST ARGUM 774
     1ENT+++++')                                                        775
 2560 LENGTH=KOINO-PI-1                                                 776
      WRITE (PRNTER,2570) (NEU(KOINO-M1),M1=1,LENGTH)                   777
 2570 FORMAT(' +++++',69A1,'+++++')                                     778
 2580 NONAME=0                                                          779
      PTSTOR=1                                                          780
      FIRST=1                                                           781
      AR=1                                                              782
      GO TO 790                                                         783
 2590 WRITE(PRNTER ,2600)                                               784
 2600 FORMAT(' +++++ THERE IS NOT SUCH A FORM IN STORE+++++')           785
      GO TO 2560                                                        786
 2610 PTSTR1=PTSTOR+1                                                   787
      KOINO3=KOINO-3                                                    788
      IF (NONAME.EQ.0) GO TO 2630                                       789
C WHEN THERE IS A FORM WITH NULL NAME SEARCH ONLY FOR NONAME=0 IN FSTORE 790
      DO  2620 N=1,NONAME                                               791
      IF (FSTORE(PTSTR1+N).NE.NEU(KOINO3-N))  GO  TO  2530              792
 2620  CONTINUE                                                         793
C AFTER  MATCH OF NAME SEARCH FOR MATCHING ARGUMENTS                    794
 2630 F2=PTSTOR+NONAME+2                                                795
      FARX=F2+2                                                         796
      FTEL=PTSTOR+FSTORE(PTSTOR)-1                                      797
      IF(FARX.EQ.(FTEL+1)) GO TO 2800                                   798
```

```
C FARX POINTER TO THE FIRST CHARACTER OF THE FORM,FTEL TO THE LAST AND F2          799
C CELL HOLDING THE NUMBER OF MARKERS                                              800
      IF (FSTORE(F2).EQ.0) GO TO 2780                                             801
      N5=FSTORE(F2)                                                               802
C N5 SHOWS THE NUMBER OF MARKERS OF THAT FORM                                     803
      DO 2750 LL=FARX,FTEL                                                        804
      DO 2650 MM=1,N5                                                             805
      TN5=MM                                                                      806
C TN5 TEMPORARY PLACE TO STORE THE ORIGINAL VALUE OF THE MARKER                   807
      MARKER=MEMPTY-MM                                                            808
      IF (FSTORE(LL).EQ.MARKER) GO TO 2640                                        809
      GO TO 2650                                                                  810
 2640 DMAR=1                                                                      811
C THIS SWITCH IS SET TO 1 WHEN THE CHARACTER EXAMINED IS A MARKER                 812
      LINE(AR)=AMPERS                                                             813
C THE AMPERSAND SHOWS THE CHARACTER IS A MARKER ITS ORDINAL VALUE FOLLOWS         814
      GO TO 2660                                                                  815
 2650 CONTINUE                                                                    816
      LINE(AR)=FSTORE(LL)                                                         817
 2660 IF (AR.EQ.79) GO TO 2680                                                    818
      AR=AR+1                                                                     819
 2670 IF (DMAR.EQ.1) GO TO 2710                                                   820
      GO  TO  2750                                                                821
C DMAR IS SET TO 1 WHENEVER THE EXAMINED CHARACTER IS A MARKER                    822
 2680 WRITE (PRNTER,2690) (LINE(NN),NN=1,AR)                                      823
 2690 FORMAT(' ',79A1)                                                            824
      IF(AR.NE.79) GO TO 2700                                                     825
      AR=1                                                                        826
      GO TO 2670                                                                  827
 2700 IF (DMAR.EQ.1) GO TO 2710                                                   828
C OTHERWISE SCANNING OF THAT FORM FINISHED                                        829
      GO TO 2580                                                                  830
 2710 IF (FIRST.NE.1) GO TO 2740                                                  831
C FIRST=1 FIRST TIME AFTER MOVING THE AMPERSAND,IMMEDIATELY CHANGES TO 0          832
      FIRST=0                                                                     833
C CONVERT THE ORDINAL VALUE INTO A CHARACTER ARRAY                                834
      DO 2720 II=1,5                                                              835
C THE VALUE CAN HAVE 5 DIGITS AT MOST SINCE INTEGER*2                             836
```

```
      NCREM=II                                                          837
      PHL=TN5/10                                                        838
      REM=TN5-PHL*10                                                    839
      ARREM(II)=NUMBER(REM+1)                                           840
      IF (PHL.EQ.0)  GO TO 2730                                         841
 2720 TN5=PHL                                                           842
 2730 LINE(AR)=ARREM(NCREM)                                             843
      GO TO 2660                                                        844
 2740 NCREM=NOREM-1                                                     845
      IF (NOREM.NE.0) GO TO 2730                                        846
      FIRST=1                                                           847
      DMAR=0                                                            848
 2750 CONTINUE                                                          849
      IF(AR.EQ.1) GO TO 2580                                            850
      AR=AR-1                                                           851
C 1 HAS BEEN ADDED TO AR WHEN  LL=FTEL                                  852
 2760 WRITE (PRNTER,2770) (LINE(NN),NN=1,AR)                            853
 2770 FORMAT(' ',79A1)                                                  854
      GO TO 2580                                                        855
C THERE ARE NO MARKERS IN THAT FORM ITS VALUE IS COPIED FROM STORE      856
 2780 WRITE (PRNTER,2790) (FSTORE (MM),MM=FARX,FTEL)                    857
 2790 FORMAT (' ',79A1)                                                 858
      GO TO 2580                                                        859
 2800 WRITE(PRNTER,2810)                                                860
 2810 FORMAT(' +++++NULL VALUE FORM+++++')                              861
      GO TO 2560                                                        862
C*************************************************************************863
C     SUBROUTINE EQ                                                     864
 2815 IF(SF.EQ.1)  GO  TO  2820                                         865
      KOINC=PTNEU                                                       866
      GO TO 2830                                                        867
 2820 KOINC=PTACT                                                       868
 2830 LOC4=KOINC-3                                                      869
      COM1=LOC4                                                         870
      IF (NEU(LOC4).NE.MARCOM)  GO TO 3000                              871
      DO 2850  II=1,4                                                   872
C TRY TO FIND HOW MANY ARGUMENTS AND CHAR PER ARG THE FUNCT HAS         873
 2840 LOC4=LOC4-1                                                       874
```

```
      IF (LOC4.LE.PI)  GO TO 2860                                          875
      IF  (NEU(LOC4).EQ.MARCOM)  GO TO 2850                                876
      NOCHA(II)=NOCHA(II)+1                                                877
      GO TO 2840                                                           878
 2850 NCOM=NCOM+1                                                          879
C KEEP A COUNTER FOR THE NO OF COMMAS INITIALLY 1,AFTER EXIT OF THE DO     880
C LOOP SHOULD BE NCOM=4                                                    881
 2860 IF  (NOCOM.NE.4) GO TO 3000                                          882
      COM2=COM1-NOCHA(1)-1                                                 883
C COMPARE 2ND,3RD ARG CHAR BY CHAR.FIRST NO OF CHARS                       884
      IF (NOCHA(1).EQ.NOCHA(2))  GO TO 2880                                885
C VALUE OF THE FUNCTION IS THE 5TH ARGUMENT                                886
 2870 BEGVAL=COM2-NOCHA(2)-NOCHA(3)-3                                      887
      ENDVAL=BEGVAL-NOCHA(4)+1                                             888
      GO TO 2910                                                           889
C COMPARE 2ND AND 3RD ARGUMENTS CHAR BY CHAR TO FIND IF EQUAL              890
 2880 NOCHA1=NOCHA(1)                                                      891
      IF (NOCHA1.EQ.0) GO TO 2900                                          892
      DO 2890 JJ=1,NOCHA1                                                  893
      IF (NEU(COM1-JJ).NE.NEU(COM2-JJ))  GO TO 2870                        894
 2890 CONTINUE                                                             895
C VALUE OF THE FUNCTION IS THE 4TH ARGUMENT                                896
 2900 BEGVAL=COM2-NOCHA(2)-2                                               897
      ENDVAL=BEGVAL-NOCHA(3)+1                                             898
 2910 L1=BEGVAL-ENDVAL+1                                                   899
      IF (L1.EQ.0) GO TO 3030                                              900
C WHEN EQ FUNCTION HAS NULL VALUE SAME PROCEDURE WITH NULL-VALUED FUNCTI   901
      IF (SF.EQ.1) GO TO 2930                                              902
C NEUTRAL FUNCTION.DELETE BY POINTING TO PREVIOUS ONE AND PI TO THE BEGN   903
      PI=PTNEU                                                             904
      PTNEU=NEU(PI)                                                        905
      DO 2920 KK=1,L1                                                      906
      IF(META.EQ.PI) CALL CLEARP                                           907
      IF(FULL.EQ.1) GO TO 3095                                             908
      NEU(PI)=NEU(BEGVAL)                                                  909
      BEGVAL=BEGVAL-1                                                      910
 2920 PI=PI-1                                                              911
      GO TO 3050                                                           912
```

```
C ACTIVE FUNCTION DELETE BY POINTING THE PREVIOUS AND PI THE BEGN                913
 2930 PI=PTACT                                                                   914
      PTACT=NEU(PI)                                                              915
      DIF=META-SPT+1                                                             916
      TDIF=DIF                                                                   917
      IF ((L1+DIF).EQ.META) GO TO 2960                                           918
      IF ((L1+DIF).LT.META)  GO TO 2980                                          919
      IF ((L1+DIF).GE.PI)  GO TO 3080                                            920
C THE UNEVALUATED ACTIVE STRING IS MOVED TO THE RIGHT FROM RIGHTMOST CHA         921
 2940 CHAR(L1+DIF )=CHAR(META)                                                   922
      META=META-1                                                                923
      DIF=DIF-1                                                                  924
      IF (META.GE.SPT) GO TO 2940                                                925
 2950 META=L1+TDIF                                                               926
 2960 SPT=1                                                                      927
C THE VALUE OF THE ACTIVE FUNCTION IS MOVED CHAR BY CHAR TO THE ACT STRING       928
      DO 2970  LL=1,L1                                                           929
      CHAR(SPT)=NEU(BEGVAL)                                                      930
      BEGVAL=BEGVAL-1                                                            931
 2970 SPT=SPT+1                                                                  932
      SPT=1                                                                      933
C THIS VALUE IS TO BE SCANNED AGAIN FOR EVALUATION,RETURN TO INITIALIZE          934
      GO TO 3050                                                                 935
 2980 DIF1=1                                                                     936
 2990 CHAR(L1+DIF1)=CHAR(SPT)                                                    937
C  THE UNEVALUATED  ACTIVE STRING IS MOVED TO THE LEFT FROM LEFTMOST CHAR.       938
      DIF1=DIF1+1                                                                939
      IF (DIF1.GT.DIF)  GO TO 2950                                               940
      SPT=SPT+1                                                                  941
      GO TO 2990                                                                 942
 3000 WRITE (PRNTER,3010)                                                        943
 3010 FORMAT(' +++++ERROR FUNCTION NOTHING HAPPENED *****TRY AGAIN+++++'         944
     1)                                                                          945
      LENGTH=KOINC-PI-1                                                          946
      WRITE (PRNTER,3020) (NEU(KOINC-M1),M1=1,LENGTH)                            947
 3020 FORMAT(' +++++',69A1,'+++++')                                              948
 3030 IF (SF.EQ.1)  GO TO 3040                                                   949
      PI=PTNEU                                                                   950
```

```
      PTNEU=NEU(PI)                                                          951
      GO TO 3050                                                             952
 3040 PI=PTACT                                                               953
      PTACT=NEU(PI)                                                          954
 3050 NCCOM=1                                                                955
      DO 3060 NN=1,4                                                         956
 3060 NCCHA(NN)=0                                                            957
 3070 GO TO 3095                                                             958
 3080 WRITE (PRNTER,3090)                                                    959
 3090 FORMAT(' +++++VALUE OF ACTIVE EQ EXCEEDS LIMITS PROCESSOR FULL++++     960
     1+')                                                                    961
      FULL=1                                                                 962
 3095 IF (FULL.EQ.1) GO TO 3270                                              963
C WHEN EQ FUNCTION HAS NULL VALUE SAME PROCEDURE WITH NULL-VALUED FUNCTION   964
      GO TO  3160                                                            965
C*************************************************************************  966
 3100 WRITE(PRNTER,3110)                                                     967
 3110 FORMAT (' +++++ERROR CODED FUNCTION DELETED+++++')                     968
      DNULL1=NULL                                                            969
C THE FOLLOWING PRINTS OUT THE ERROR CODED FUNCTION                          970
 3120 IF  (PTACT.LE.PTNEU)  GO   TO   3150                                   971
      LOC2=PTNEU                                                             972
 3130 LENGTH=LOC2-PI-1                                                       973
      WRITE (PRNTER,3140) (NEU(LOC2-J2),J2=1,LENGTH)                         974
 3140 FORMAT(' +++++',69A1,'+++++')                                          975
      GO TO 790                                                              976
 3150 LOC2=PTACT                                                             977
      GO  TO  3130                                                           978
 3160 IF (SF.EQ.0) GO TO 10                                                  979
      SF=0                                                                   980
      GO TO 10                                                               981
 3170 WRITE (PRNTER,3180)                                                    982
 3180 FORMAT (' +++++TRAC JOB FINISHED THANK YOU GOOD-BYE,YEIA-XARA COST     983
     1AS+++++')                                                              984
 3190 STOP                                                                   985
 3200 WRITE (PRNTER,3210)                                                    986
 3210 FORMAT(' +++++TO END TRAC JOB ,TYPE '' IN COLUMN 1++++++')             987
      DO 3220  J=1,12                                                        988
```

```
 3220 CHAR(J)=IDLE(J)                                                    989
      META=12                                                            990
      SPT=1                                                              991
      GO TO 10                                                           992
C WHEN TRACE ON THE WHOLE FUNCTION IS COPIED FROM NEUTR STRING BEFORE EV 993
 3230 IF  (PTACT.LE.PTNEU)  GO   TO   3260                               994
      LOC2=PTNEU                                                         995
 3240 LENGTH=LOC2-PI-1                                                   996
      WRITE (PRNTER,3250) (NEU(LOC2-J2),J2=1,LENGTH)                     997
 3250 FORMAT (' ',79A1)                                                  998
      GO TO 240                                                          999
 3260 LOC2=PTACT                                                        1000
      GO TO 3240                                                        1001
 3270 FULL=0                                                            1002
C FULL TRAC PROCESSOR IS PRINTED OUT AND THE IDLING PROC. IS RELOADED   1003
      WRITE (PRNTER,3280) CHAR                                          1004
 3280 FORMAT(' +++++',60A1,'+++++')                                     1005
      GO TO 3190                                                        1006
      END                                                               1007
C***********************************************************************1008
      SUBROUTINE CLEARP                                                 1009
      IMPLICIT  INTEGER*2  (A-Z)                                        1010
      COMMON /PARAM/PARENL,PARENR,COMMA,SHARP,BLANK ,QUOTE,AMPERS       1011
      COMMON/CONST/BEGP,SPT,PI,META,MEMPTY,FORGET,MARCOM,TOTAL,TRACE    1012
     1,MRQUOT,CARD                                                      1013
      COMMON /STORE/ CHAR(1000),VAL(80),FSTORE(1000),LINE(80),NUMBER(10)1014
      COMMON/NEUPOI/PTACT,PTNEU,ENDP,L1,SF,FULL                         1015
      COMMON /FUNCT/FA(11),FB(11),FC(11)                                1016
      DIMENSION  NEU(1000)                                              1017
      EQUIVALENCE  (CHAR,NEU)                                           1018
      COMMON /INPOUT/ READER,PRNTER,IDLE,TIMES                          1019
      INTEGER*4  READER,PRNTER,IDLE*2(12),TIMES*2                       1020
      COMMON/LETTER/A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z 1021
      IF (SPT.EQ.1)  GO  TO  40                                         1022
      NMETA=META-SPT+1                                                  1023
      TSPT=1                                                            1024
 10   CHAR(TSPT)=CHAR(SPT)                                              1025
      SPT=SPT+1                                                         1026
```

```
      IF (SPT.GT.META) GO TO 20                                          1027
      TSPT=TSPT+1                                                        1028
      GO TO 10                                                           1029
20    META=NMETA                                                         1030
      SPT=1                                                              1031
      WRITE (PRNTER,30)  META,PI                                         1032
30    FORMAT(' +++++NEW VALUE OF META=',I4,10X,'PI=',I4,'++++++')        1033
      GO TO 60                                                           1034
40    WRITE (PRNTER,50) PI                                               1035
50    FORMAT(' +++++PROCESSOR FULL,EXECUTION STOPS+++++'/' +++++PI=',I4, 1036
     1'++++++')                                                          1037
      FULL=1                                                             1038
60    RETURN                                                             1039
      END                                                                1040
```